Parallel Systems Architecture Lab

Babak Falsafi

Team: Nora Abi Akar, Alexandros Daglis, Mario Drumond, Damien Hilloulin, Ivo Mihailovic, Nooshin Mirzadeh, Stanko Novakovic, Javier Picorel, Arash Pourhabibi, Dmitrii Ustiugov

Server Benchmarking with CloudSuite 3.0

PARSA, EPFL

EuroSys'16, London, UK

Preface

- CloudSuite: Benchmark suite of cloud services
- Docker: Automates application deployment via containers
- PerfKit: Automates benchmarking of cloud providers
- QFlex: Quick& Flexible Rack-Scale Architectural Simulator

- The tutorial is interactive
  - Please ask questions anytime during tutorial

Agenda

CloudSuite 3.0 benchmarks overview

CloudSuite 3.0 on real hardware

Full-system simulation with QFlex

CloudSuite 3.0:
A Suite for Emerging Scale-out Applications

Mario Drumond

A Brief History of IT

Mobile Era

Consumer Era

1970s- 1980s 1990s Today+

Mainframes

PC Era

- From computing-centric to data-centric
- Consumer Era: Internet-of-Things in the Cloud

Data is Shaping Future of IT

If the Digital Universe were represented by the memory in a stack of tablets, in **2013** it would have stretched two-thirds the way to the Moon*

By **2020**, there would be 6.6 stacks from the Earth to the Moon*

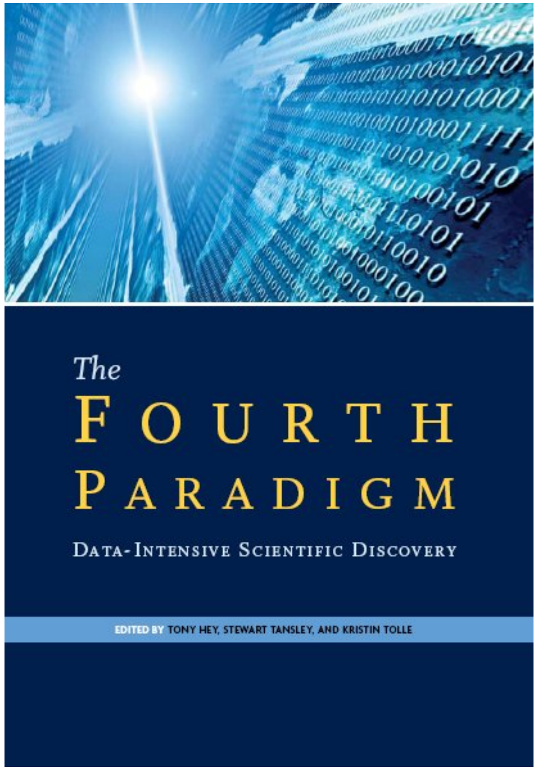
Source: IDC, 2014
• iPad Air – 0.29" thick, 128 GB

- Data growth (by 2015) = 100x in ten years [IDC 2012]
  - Population growth = 10% in ten years
- Monetizing data for commerce, health, science, services, ….
- Big Data is shaping IT & pretty much whatever we do!

Data Shaping All Science & Technology

Science (traditionally HPC) entering 4th paradigm

- Analytics using IT on
    - Instrument data
    - Simulation data
    - Sensor data
    - Human data
    - …



The
FOURTH
PARADIGM

DATA-INTENSIVE SCIENTIFIC DISCOVERY

EDITED BY TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE

Complements theory, empirical science & simulation

HPC & data-centric cloud services are converging

Modern HPC in the Datacenter

- Increasing popularity of analytics workloads
  - Closely related to traditional HPC workloads (e.g., graphs)
- Service providers don't acquire supercomputers
  - All workloads share the same datacenter
  - Cost hard to sustain (e.g., IBM discontinued BlueGene)
- HPC is taking a turn towards datacenters
  - Datacenter provides higher availability, lower queue times, flexibility
  - E.g., Amazon provides HPC instances
  - E.g., Genomic analysis with Hadoop [biodatomics]

Datacenters are the heart of both cloud services and science

Datacenters Growing Rapidly

Source: James Hamilton, 2012

Each day Amazon Web Services adds enough new capacity to support all of Amazon.com's global infrastructure through the company's first 5 years, when it was a $2.76B annual revenue enterprise

Daily growth in 2012 = First five years of business!

How are we Designing Cloud Systems?

- Spoiler alert: We are doing it wrong!
- Modern servers based on desktop processors
- Server design guided by unrepresentative benchmarks

Design needs to be driven by cloud-representative benchmarks

Traditional Benchmarks

- SPEC, PARSEC, TPC-C, SPLASH, …
  - Single machine metrics (e.g., SPEC score)
  - Metrics related to the performance of a single component (usually CPU)
  - Vastly different application footprints

TPC Transaction Processing Performance Council

- None of these run on a datacenter

PARSEC ['pär-''sek] A unit of measure

spec®

Traditional benchmarks not suitable for cloud evaluation and research

What traditional benchmarks miss?

- No notion of end-to-end performance metrics
  - Traditional metrics do not reflect user experience

- Cloud services have extensive instruction footprint
  - Multi-megabyte instruction working sets
  - Overall performance highly dependent on processor's frontend

- Cloud services deal with big data
  - Datasets do not fit in on-chip caches

Whatever those benchmarks are modeling, does not apply here

Cloud Service Requirements

- Throughput:
  Owners want computing power for their money

- Latency (online services):
  Users abandon services if response time is high
  - Amazon: 100ms of latency can cause 1% of sales loss
  - YouTube: Users start abandoning video after 2 seconds of wait

CloudSuite Benchmark Suite

- CloudSuite's goal:
  Assess performance of cloud services on modern hardware
  - Make the case for cloud service representativeness
  - Identify improvement opportunities for server hardware

- End-to-end performance metrics
  - Hard problem; still open research question!

Cloud Benchmarking with CloudSuite 3.0 (cloudsuite.ch)

Data Analytics
Machine learning

Graph Analytics
GraphX

In-Memory Analytics
Recommendation System

Web Search
Apache Solr & Nutch

Media Streaming
Nginx, HTTP Server

Web Serving
Nginx, PHP server

Data Caching
Memcached

Data Serving
Cassandra NoSQL

Building block for Google PerfKit, EEMBC Big Data!

Brief History of CloudSuite

PARALLEL SYSTEMS ARCHITECTURE LAB

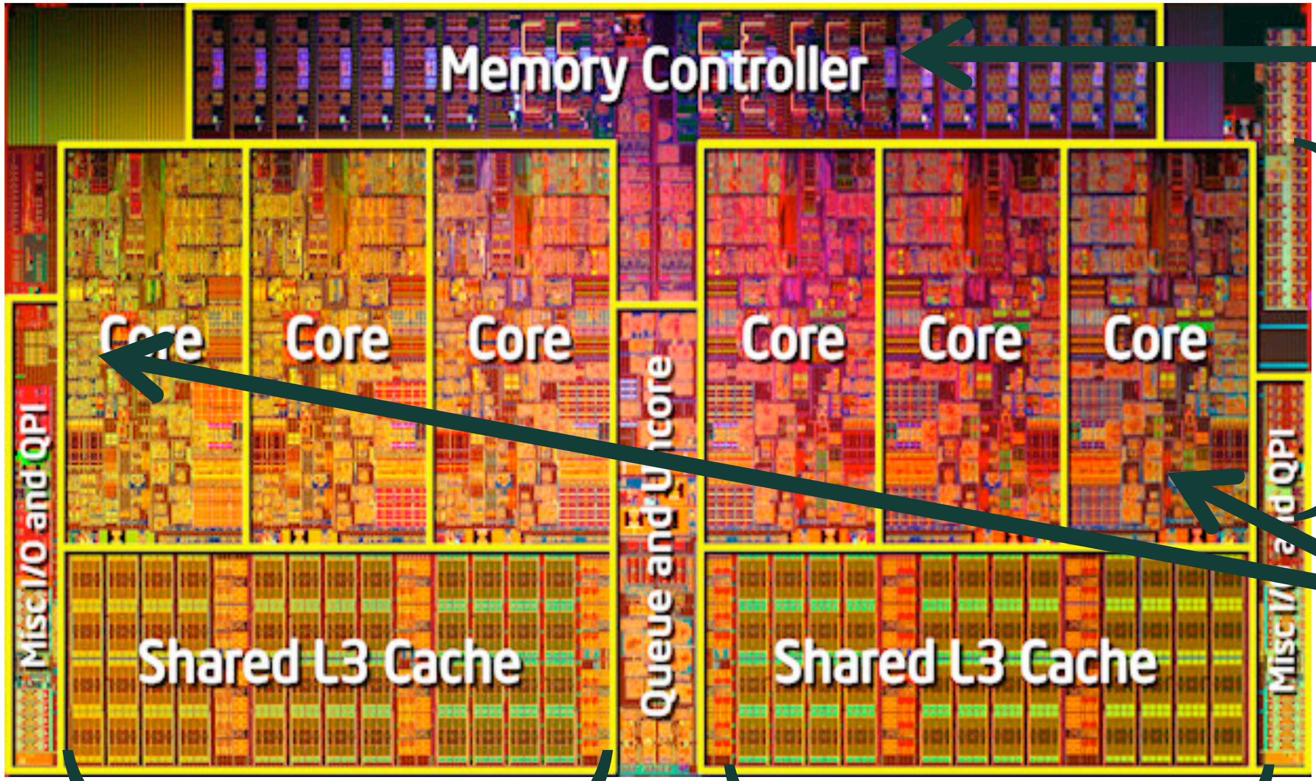
- Clearing the Clouds [Ferdman et al., ASPLOS'12] (CloudSuite 1.0)
  - Fundamental mismatch of cloud workloads and modern servers
  - Sever silicon real-estate misuse in current systems
- CloudSuite2.0 – two additional workloads
  - Graph Analytics, Data Caching
- Insights derived from CloudSuite impacted industry
  - E.g., Cavium ThunderX
- Integration with Google's PerfKit Benchmarker
- Now: Official release of **CloudSuite**

Clearing the Clouds in a Nutshell

[ASPLOS 2012]

PARALLEL SYSTEMS ARCHITECTURE LAB

Workload/Server Mismatch



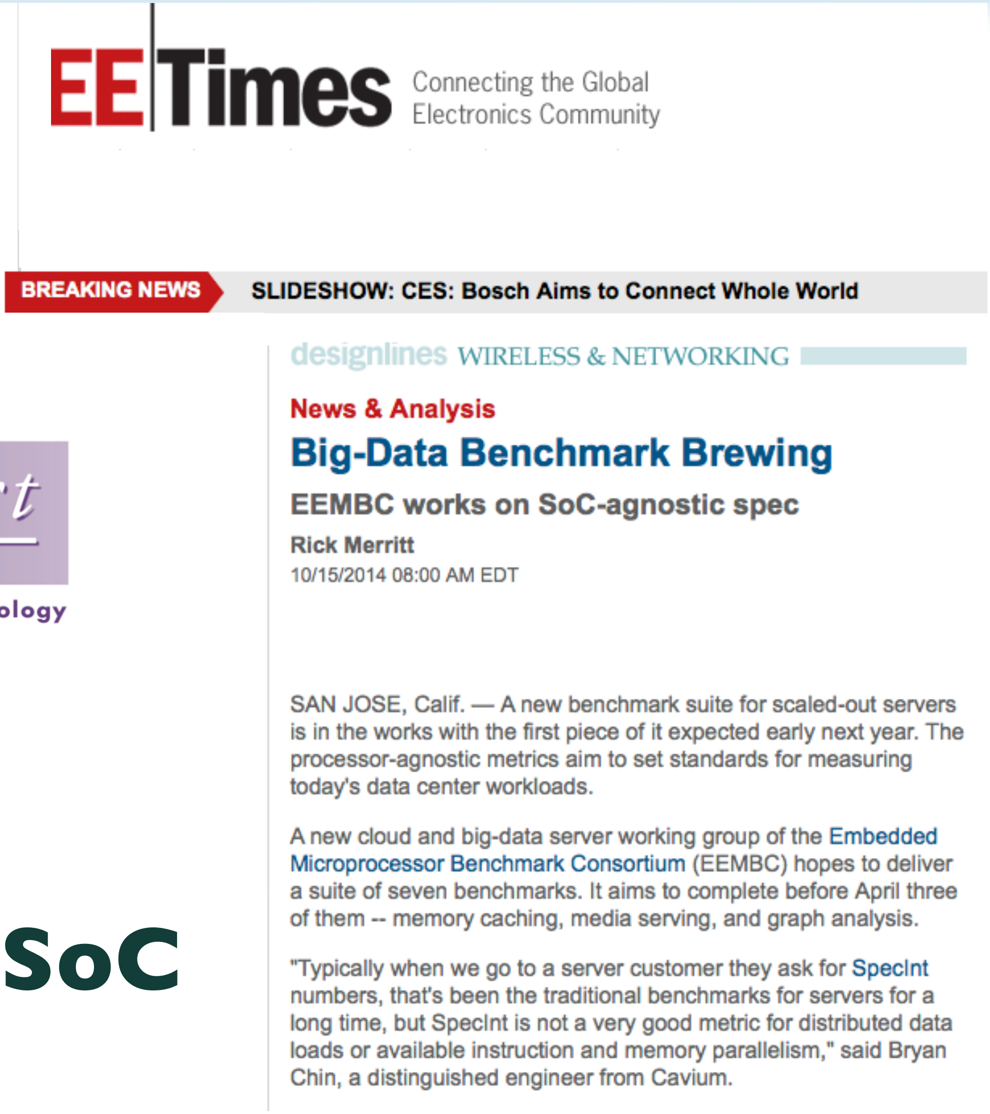
B/W unused!

Cores too fat!

Too few cores!

20 MB (80%) waste of space (no reuse)!

Cavium ThunderX
Scale-Out Processor



EE Times Connecting the Global Electronics Community

BREAKING NEWS SLIDESHOW: CES: Bosch Aims to Connect Whole World

designlines WIRELESS & NETWORKING

News & Analysis

Big-Data Benchmark Brewing

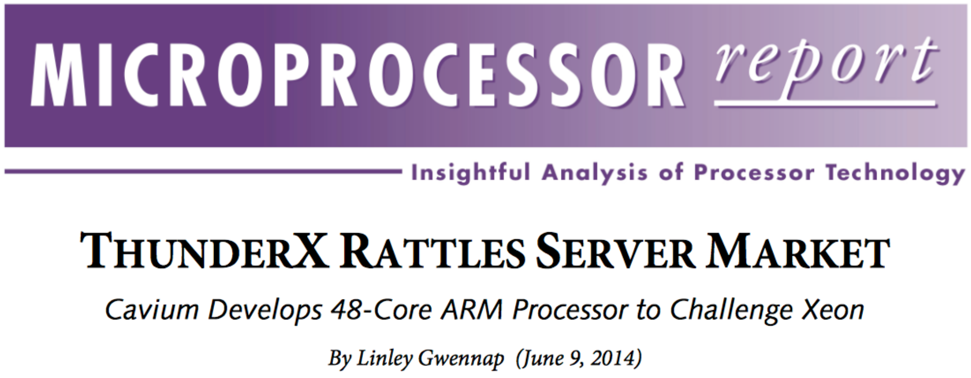
EEMBC works on SoC-agnostic spec

Rick Merritt
10/15/2014 08:00 AM EDT

SAN JOSE, Calif. — A new benchmark suite for scaled-out servers is in the works with the first piece of it expected early next year. The processor-agnostic metrics aim to set standards for measuring today's data center workloads.

A new cloud and big-data server working group of the Embedded Microprocessor Benchmark Consortium (EEMBC) hopes to deliver a suite of seven benchmarks. It aims to complete before April three of them -- memory caching, media serving, and graph analysis.

"Typically when we go to a server customer they ask for SpecInt numbers, that's been the traditional benchmarks for servers for a long time, but SpecInt is not a very good metric for distributed data loads or available instruction and memory parallelism," said Bryan Chin, a distinguished engineer from Cavium.



MICROPROCESSOR *report*

Insightful Analysis of Processor Technology

THUNDERX RATTLES SERVER MARKET

Cavium Develops 48-Core ARM Processor to Challenge Xeon

By Linley Gwennap (June 9, 2014)

48-core 64-bit ARM SoC

[blueprinted at EPFL]:

- Designed to serve data
- Specialized chip design for servers
- 10x better efficiency than Xeon

CAVIUM NETWORKS

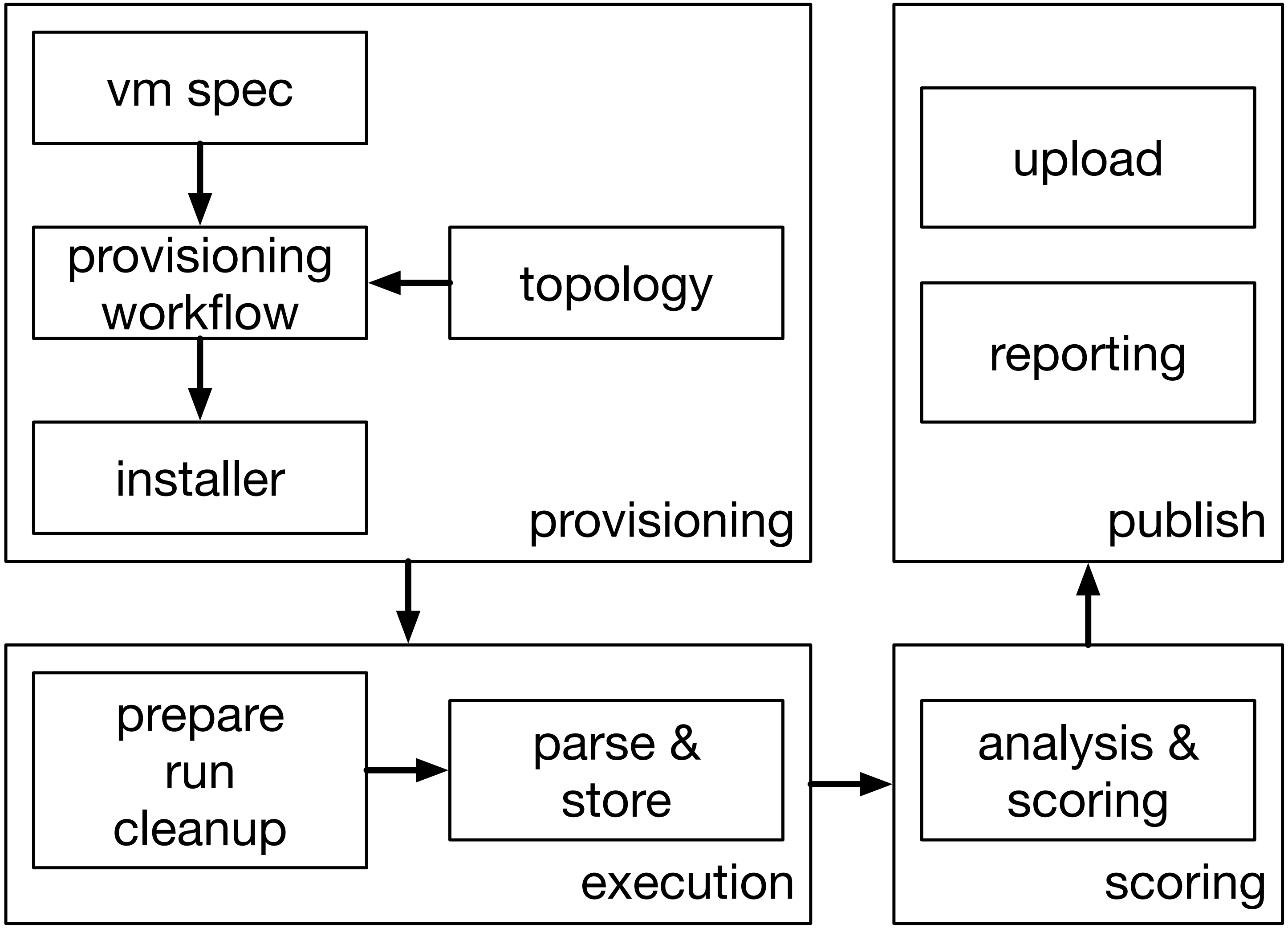
Google PerfKit Benchmarker

- Goal: Standardize Cloud performance evaluation
- A tool to compare cloud service providers
- Consortium of industry and academics

- Fully automates benchmarks including creating databases, disks, networks, and virtual machines
  - 26+ benchmarks
  - CloudSuite benchmarks included

- Shared publicly on GitHub
  - http://www.github.com/GoogleCloudPlatform/PerfKitBenchmarker

Perfkit's Workflow



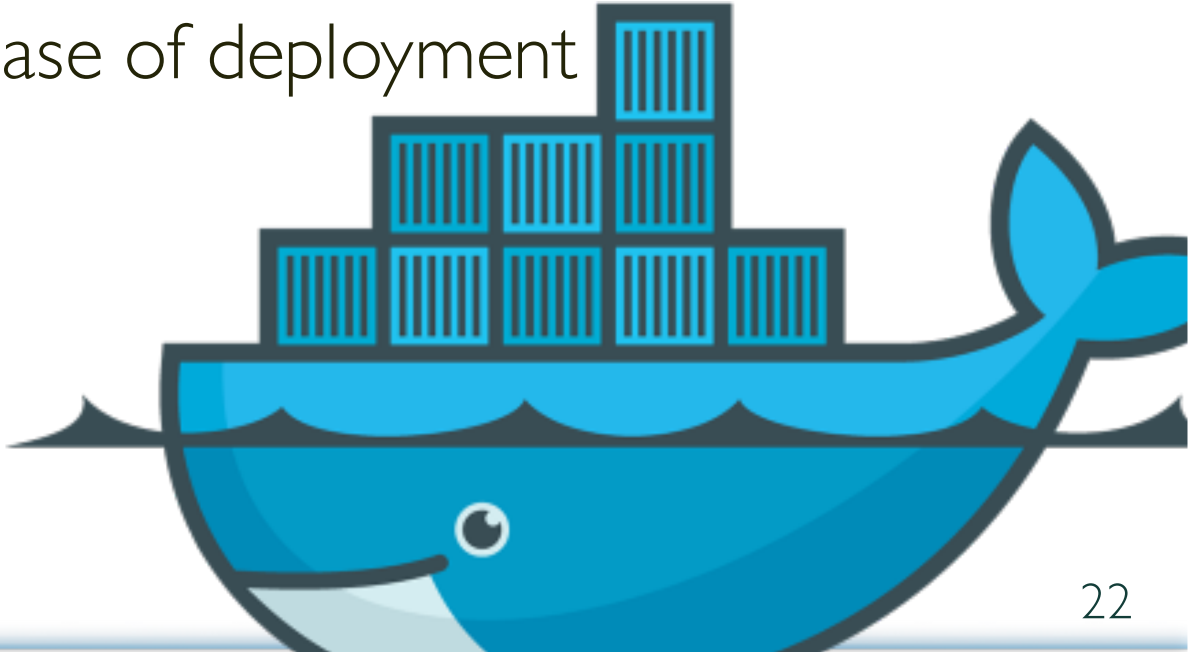
Perfkit automates the entire process

What’s new in CloudSuite 3.0

- A couple of different workloads
  - New: In-Memory Analytics
  - New software stack: Graph Analytics, Media Streaming, Web Search

- Updated software packages of all workloads

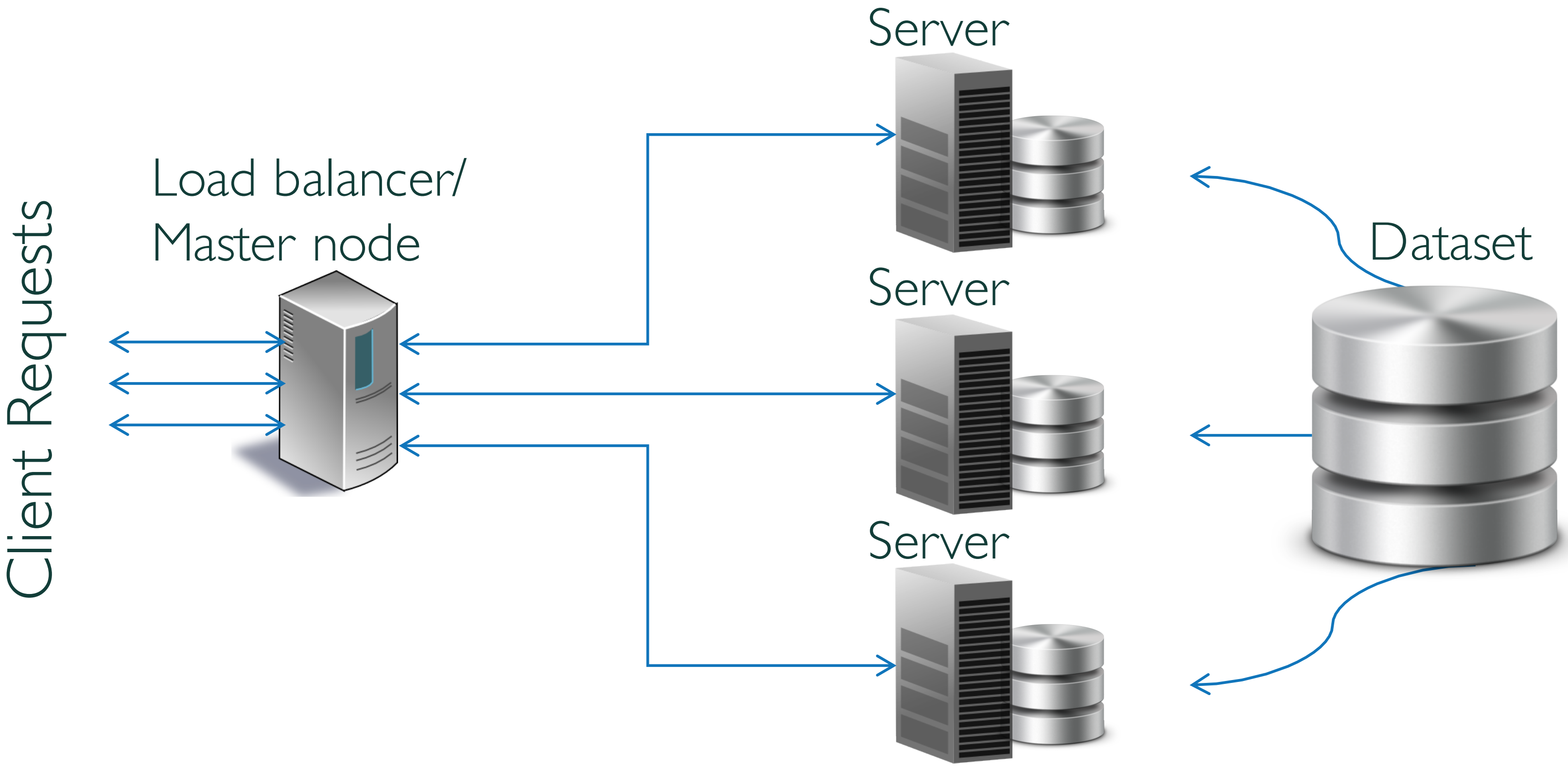
- Docker containers → ease of deployment
  - This is huge! (literally)



Target Audience

- System designers
  - Assess & compare systems' performance of cloud workloads
- Computer architects
  - Derive insights for future server design
- HPC community
  - Datacenter & HPC applications converging

Key Cloud Service Characteristics



- Serve independent requests/tasks
- Operate on huge dataset split into shards
- Communicate infrequently
- Strict real-time constraints

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Offline Benchmarks

- Operate on large datasets

- Usually a machine learning algorithm over large datasets

- Performance metric:
  - Completion time (for a given input size)

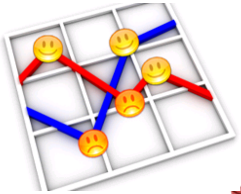
- No real-time constraints

Data Analytics

- Massive amounts of human-generated data (Big Data)
- Extract useful information from data
  - Predict user preferences, opinions, behavior
  - Benefit from information (e.g., business, security)
- Several examples
  - Book recommendation (Amazon)
  - Spyware detection (Facebook)

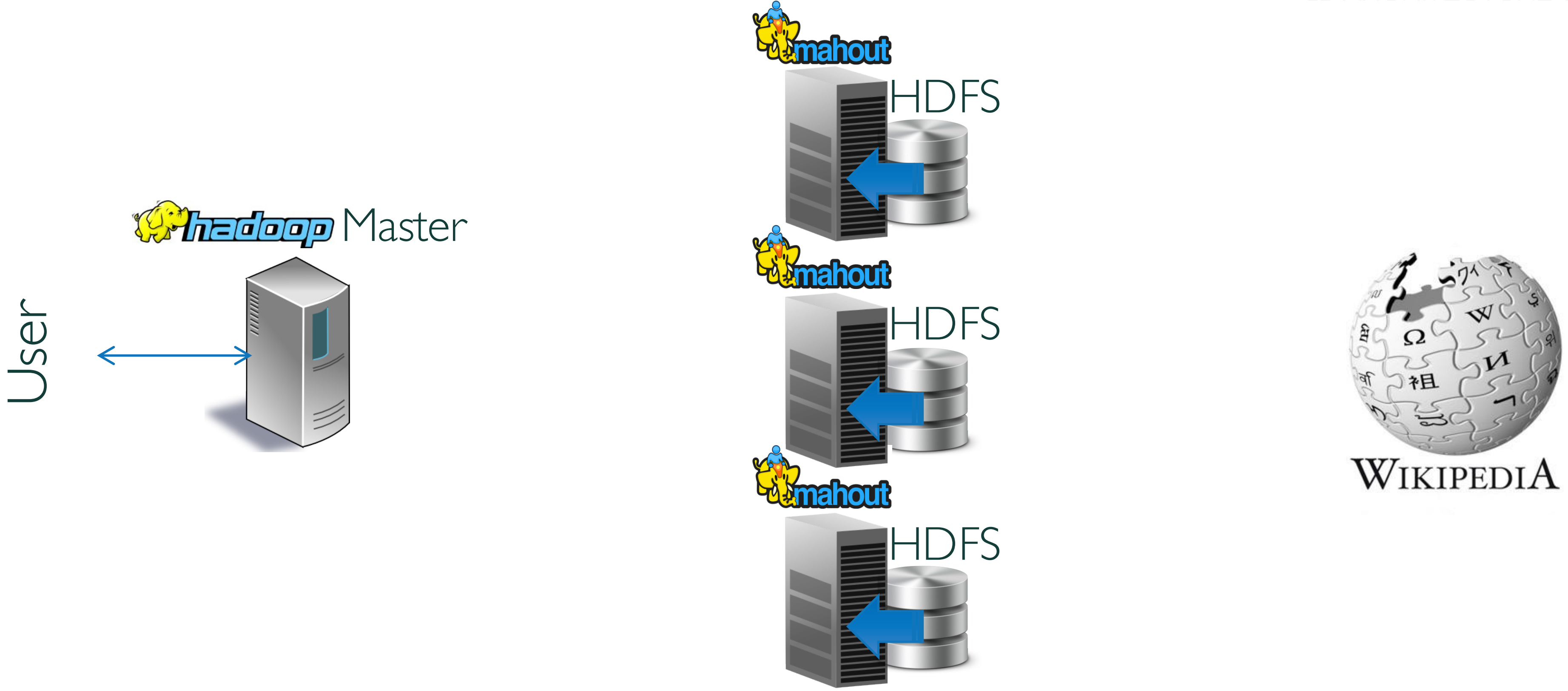
Data Analytics Benchmark

- **Application:** Text classification
  - Sentiment analysis
  - Spam Identification



- **Software:** Mahout (Apache)
  - Popular MapReduce machine learning library
- **Dataset:** Wikipedia English page articles

Data Analytics Benchmark



- Build a model from a Wikipedia training input
- Master sends Wikipedia documents for classification
- Slaves classify documents locally using model
- Slaves send results to master
- Performance metric: completion time

CloudSuite 3.0 Benchmarks

PARALLEL SYSTEMS ARCHITECTURE LAB

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

In-Memory Analytics

- In-memory processing for human-generated data

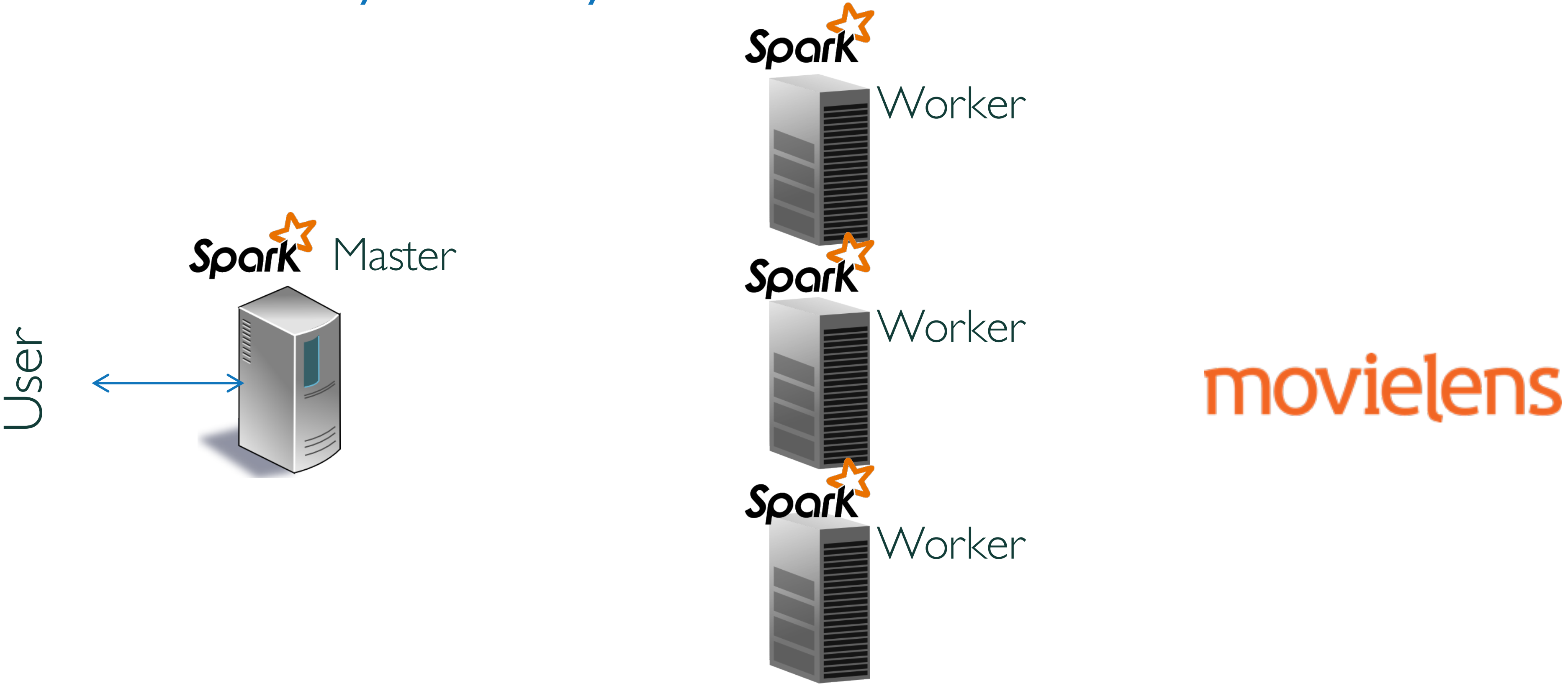
- Extract useful information from user data
  - Predict user preferences, rates

- Several examples
  - Movie recommendation (Netflix)
  - Item recommendation (Amazon)
  - Song recommendation (Spotify)
  - Recommending new friends, groups, … (Social networks)

In-Memory Analytics Benchmark

- **Application:** Collaborative filtering
  - Recommendation systems
- **Software:** Apache MLlib
  - Popular Apache Spark machine learning library
- **Dataset:** Movielens video database

In-Memory Analytics Benchmark



- Build a recommendation model with ALS matrix factorization
- Master partitions rating matrix, user & item vectors; sends them to workers
- Workers perform local matrix factorization
- Workers send results to master
- Performance metric: completion time

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

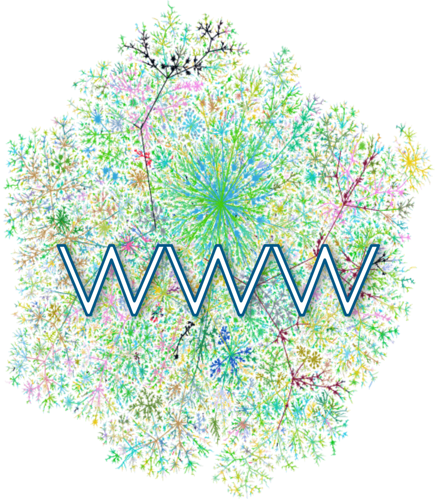
- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Graph Analytics

- Parallel distributed graph processing

- Data mining on graphs

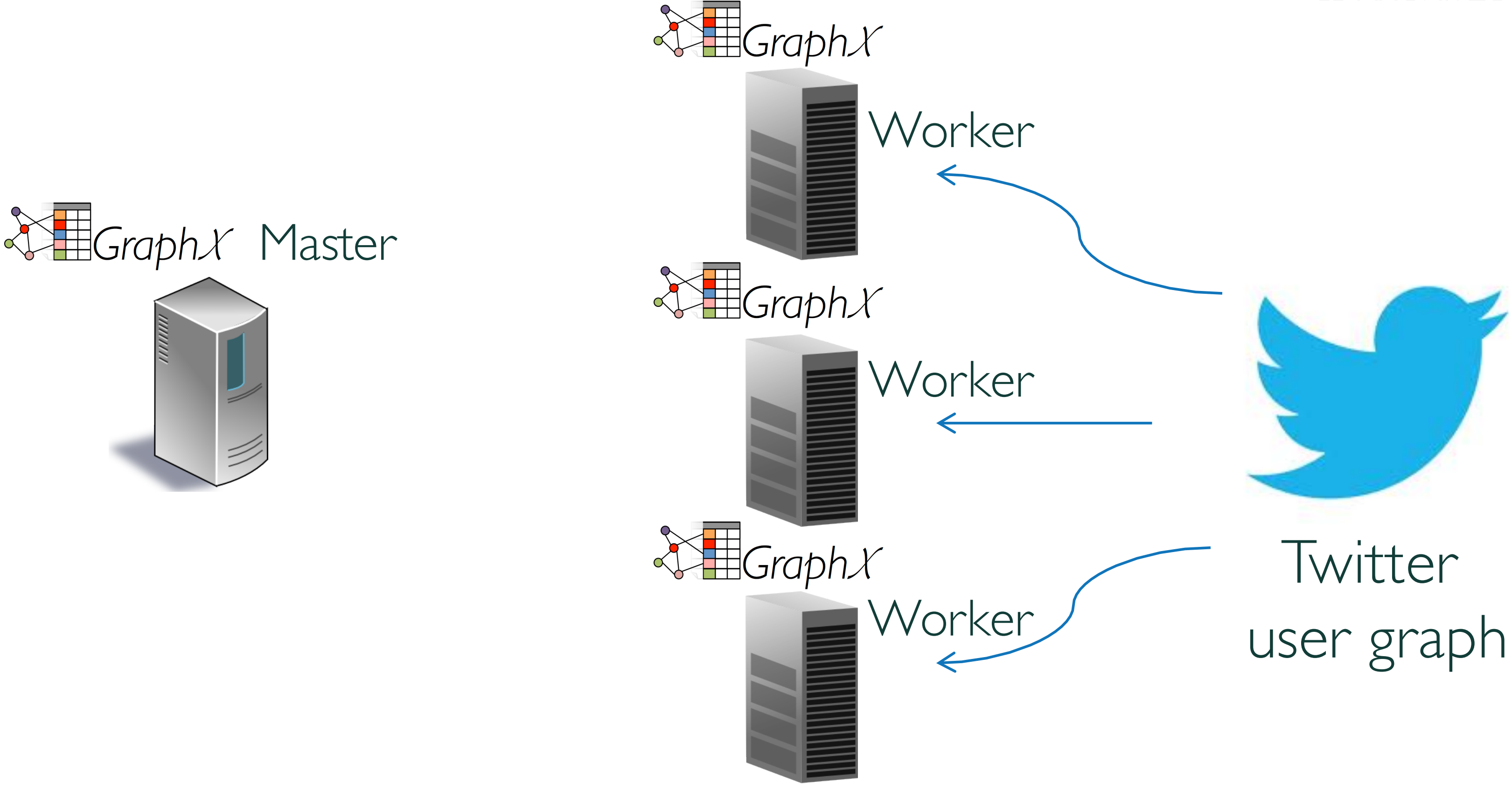
- Graph examples
  - Social networks (Facebook, Twitter)
  - Web graph



Graph Analytics Benchmark

- Application: PageRank
  - Measures influence of Twitter users
  - How much attention followers pay to a user
- Software: Apache GraphX
  - Parallel framework for graph processing
- Dataset
  - Twitter user graph

Graph Analytics Benchmark



- Distributes the graph across nodes
- Iterative computation: Always with adjacent vertices
- Communication across machines for adjacent vertices
- Output: influence of each user in the graph
- Performance metric: completion time

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Online Benchmarks

- Operate on large datasets

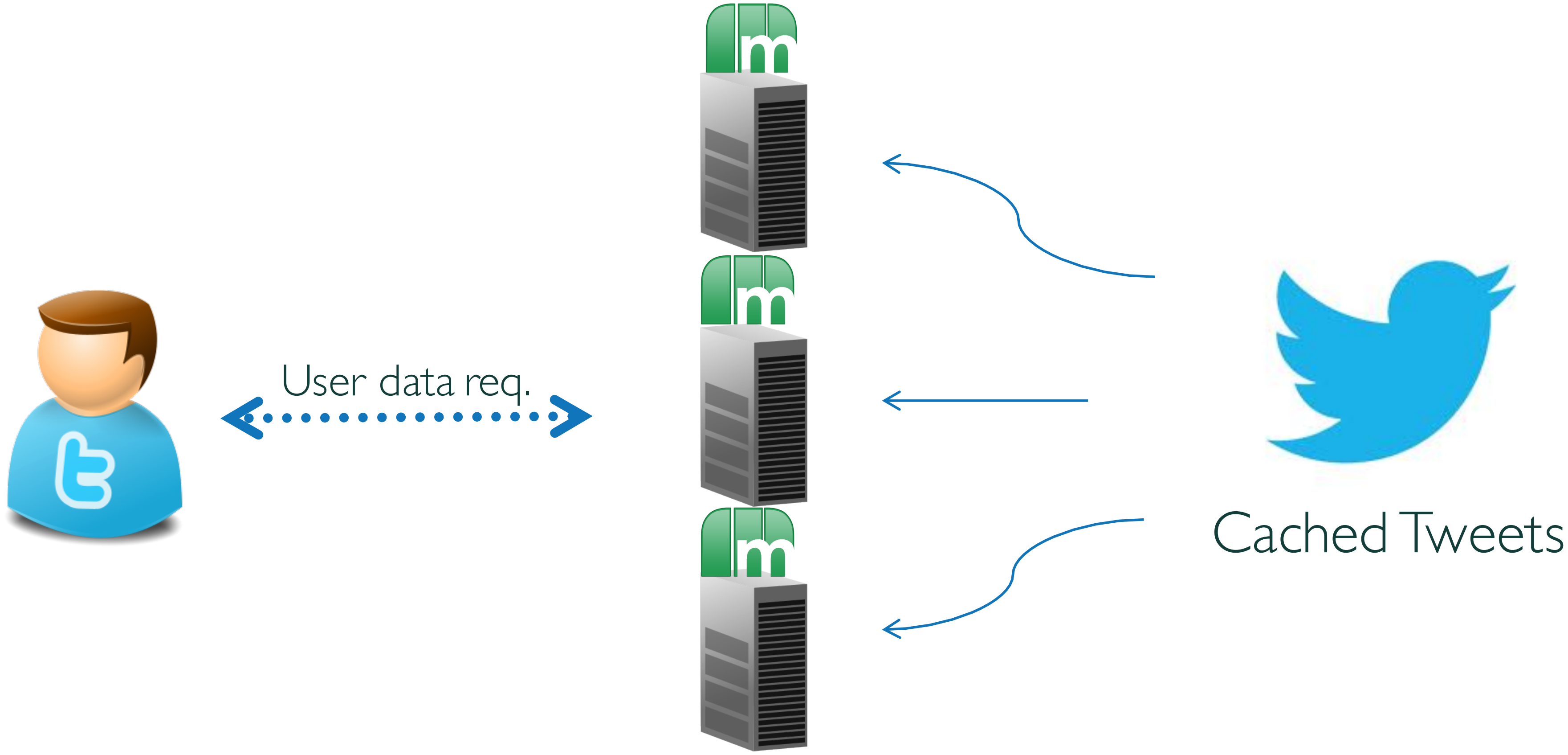
- Throughput is important, but also need high service quality
  - Tail latency of requests is critical for service quality
  - Goal: Maximize throughput *under QoS target*

- Performance metrics:
  - Throughput (metric is benchmark-specific)
  - Delivered QoS (in terms of N-th percentile latency)

Data Caching

- Web apps are latency-sensitive
- Fetching data from disk is slow
- Caching data in memory for fast data access
    - General-purpose, in-memory key-value store
    - Caches data for other apps, another tier before back-end

Data Caching Benchmark



- Driver emulates Twitter users
- Memcached software to cache data in memory
- If data not found in cache, issues a disk access request
- Performance metrics: # requests/second, N-th pct latency

CloudSuite 3.0 Benchmarks

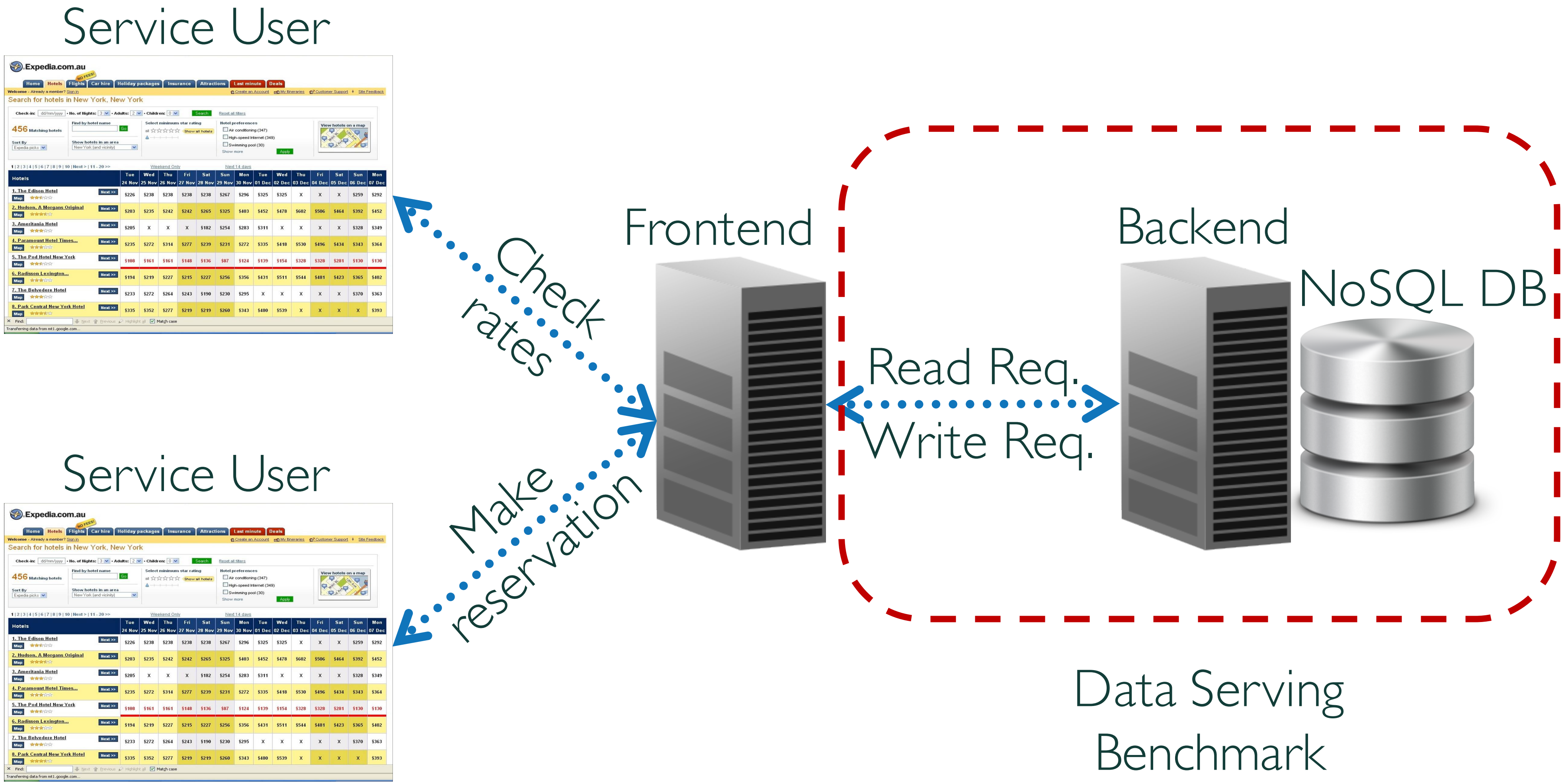
- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Data Serving

- Global-scale online services rely on NoSQL datastores
    - Inherently scalable
    - Suitable for unpredictable schema changes
- Scale out to meet service requirements
    - Accommodate fast data generation rate

Data Serving Operation



Service User

Service User

Check rates

Make reservation

Frontend

Backend

NoSQL DB

Read Req.

Write Req.

Data Serving Benchmark

Data Serving Benchmark

Request Emulator

Backend

Read & Write Requests

- Yahoo! Cloud Serving Benchmark (YCSB) driver
    - Predefined mixes of read/write operations
    - Popularity of access distributions (e.g., zipfian)
    - Interface to popular datastores (e.g., Cassandra, HBase)

Data Serving Benchmark

Request Emulator

Read & Write Requests

Backend

- Cassandra datastore
  - Popular NoSQL: many use cases (e.g., Expedia, eBay, Netflix)
- Driver generates dataset
  - Defines number & size of fields
  - Populates datastore
- Performance metrics: R/W ops/s, N-th pct latency

CloudSuite 3.0 Benchmarks

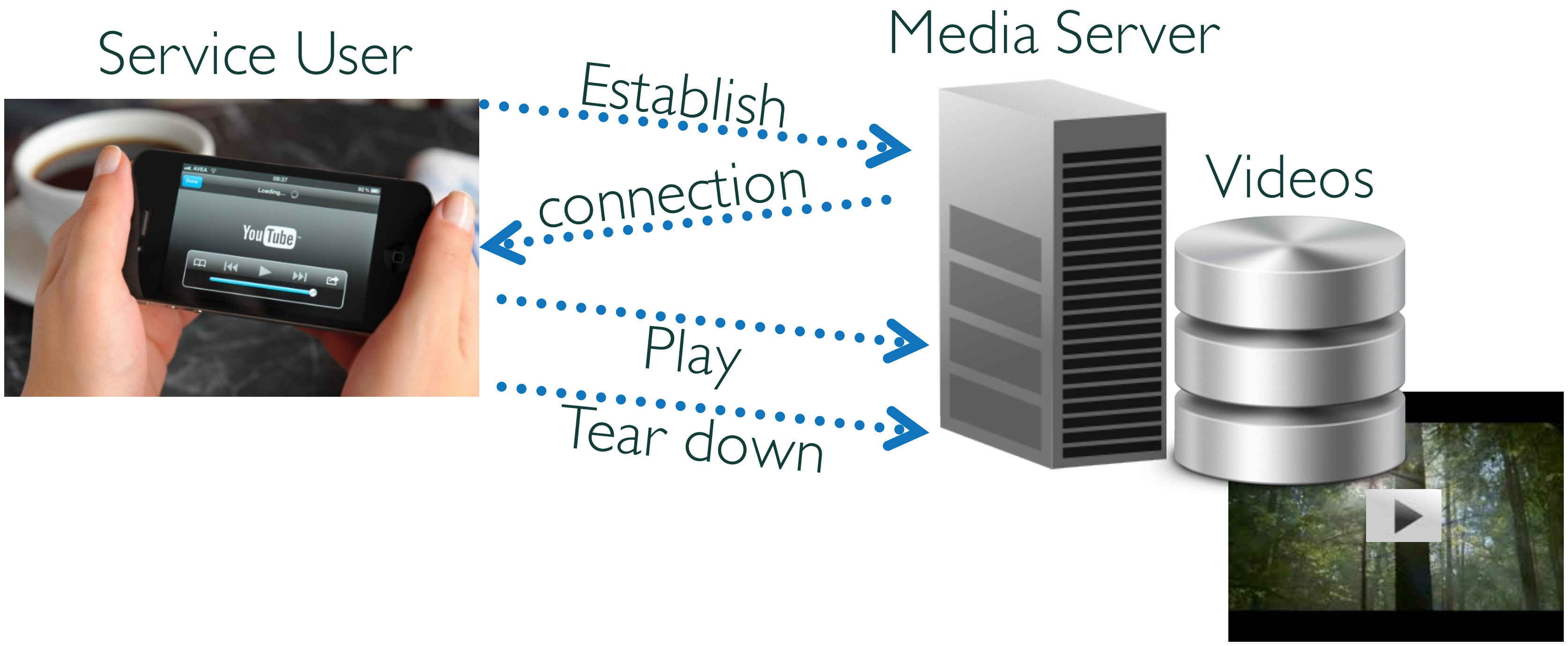
- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Media Streaming

- Media streaming expected to dominate internet traffic
- Increasing popularity of media streaming services
  - Video sharing sites, movie streaming services, etc.

Media Streaming Operation



Service User

Media Server

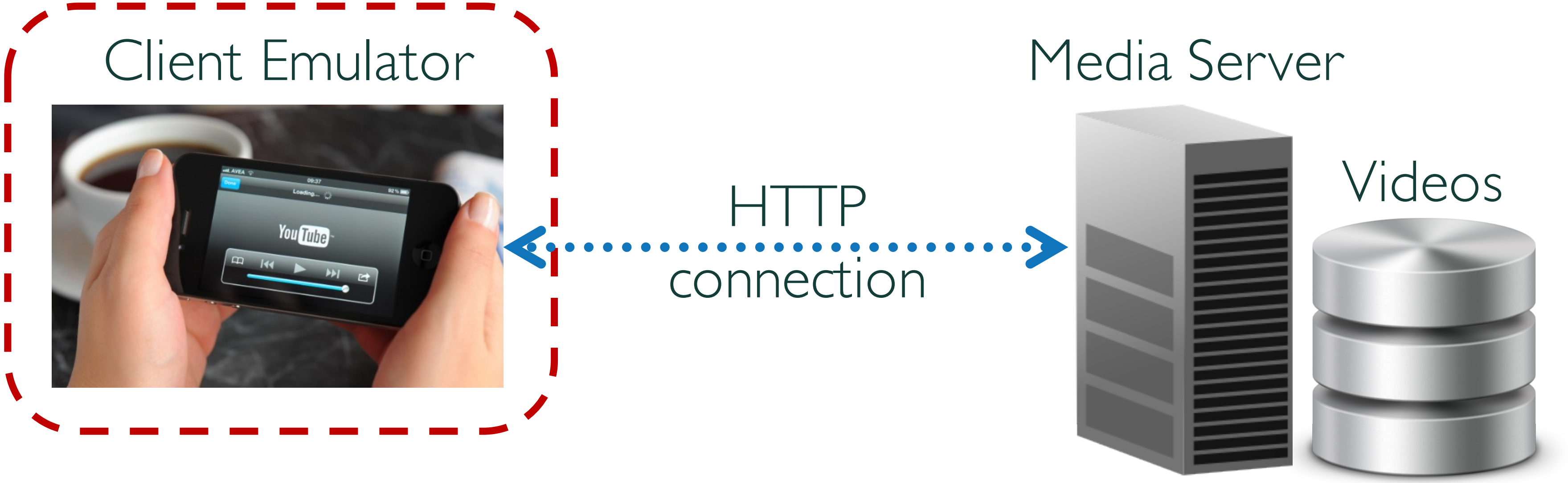
Videos

Establish connection

Play

Tear down

Media Streaming Benchmark



Client Emulator

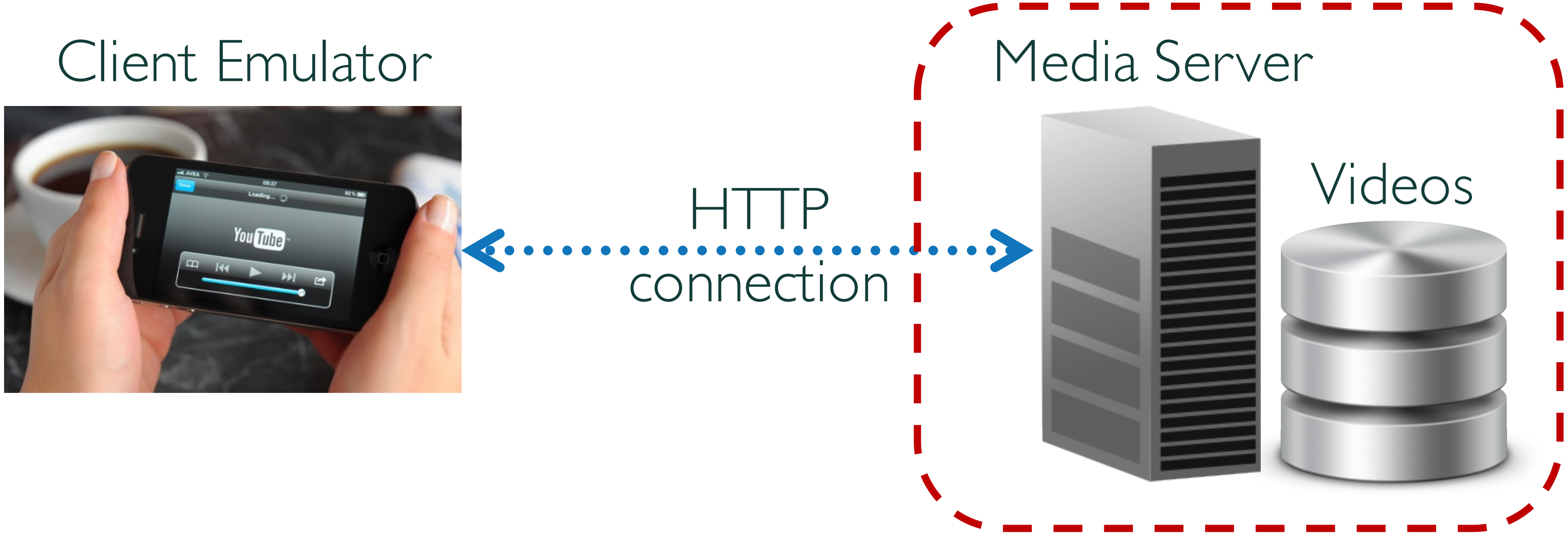
HTTP connection

Media Server

Videos

- Implements HTTP communication
- Uses the videoperf client, based on the httperf traffic generator
- Allows a flexible mix of requests
  - Different video lengths and qualities

Media Streaming Benchmark



Client Emulator

HTTP connection

Media Server

Videos

- Server required to support HTTP
  - Nginx server
- Dataset consists of a mix of pre-encoded videos
  - Four video qualities of different durations (240p, 360p, 480p, 720p)
  - Exponential popularity distribution
- Performance metrics: streaming bandwidth (Kbps), avg. reply delay

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Web Search

- Most popular online service
    - Numerous search engines deployed by industry

Google™

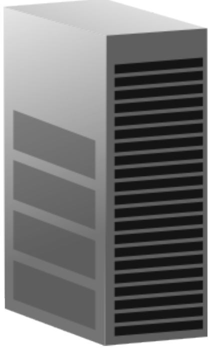
bing™

Yandex

Web Search Operation

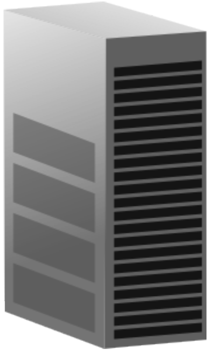
Search User

Frontend



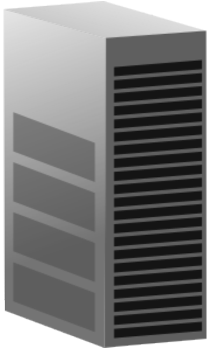
Query Term	Document
…	
Benchmark	1, 10, 17, …
CloudSuite	3, 45, …
Datacenter	9, 11, 14, 45, …
EPFL	17, 10, 15, …
PerfKit	3, 4,18
…	

Index Serving Node (ISN)



Query Term	Document
…	
Benchmark	1, 5, 7, …
CloudSuite	5, 2, …
Datacenter	7, 10, 17, 20, …
EPFL	2, 4, 6, 8, 23, …
PerfKit	3, 5, 20, 33, 34, 55, …
…	

Inverted Index



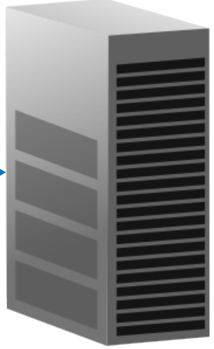
Query Term	Document
…	
Benchmark	1, 6, 19, …
CloudSuite	5, 40, …
Datacenter	6, 10, 13, 20, …
EPFL	5, 10, 23, …
PerfKit	3, 6, 10, 20, …
…	

Web Search Operation

Search User

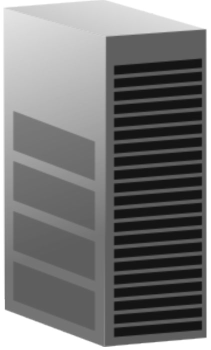
Query = “EPFL”

Frontend



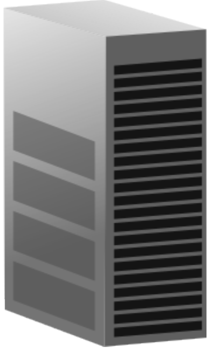
Query Term	Document
...	
Benchmark	1, 10, 17, ...
CloudSuite	3, 45, ...
Datacenter	9, 11, 14, 45, ...
EPFL	17, 10, 15, ...
PerfKit	3, 4,18
...	

Index Serving Node (ISN)



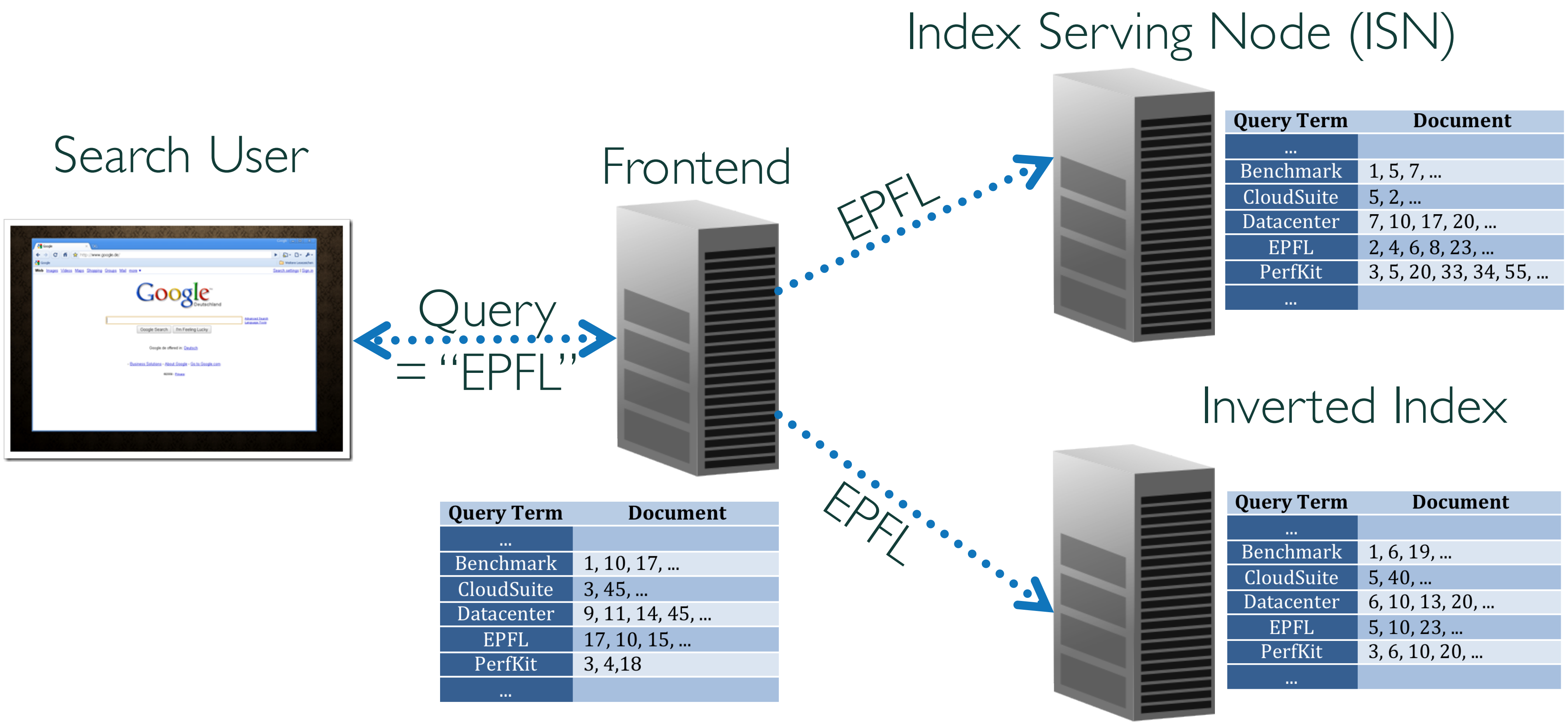
Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Inverted Index



Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

Web Search Operation

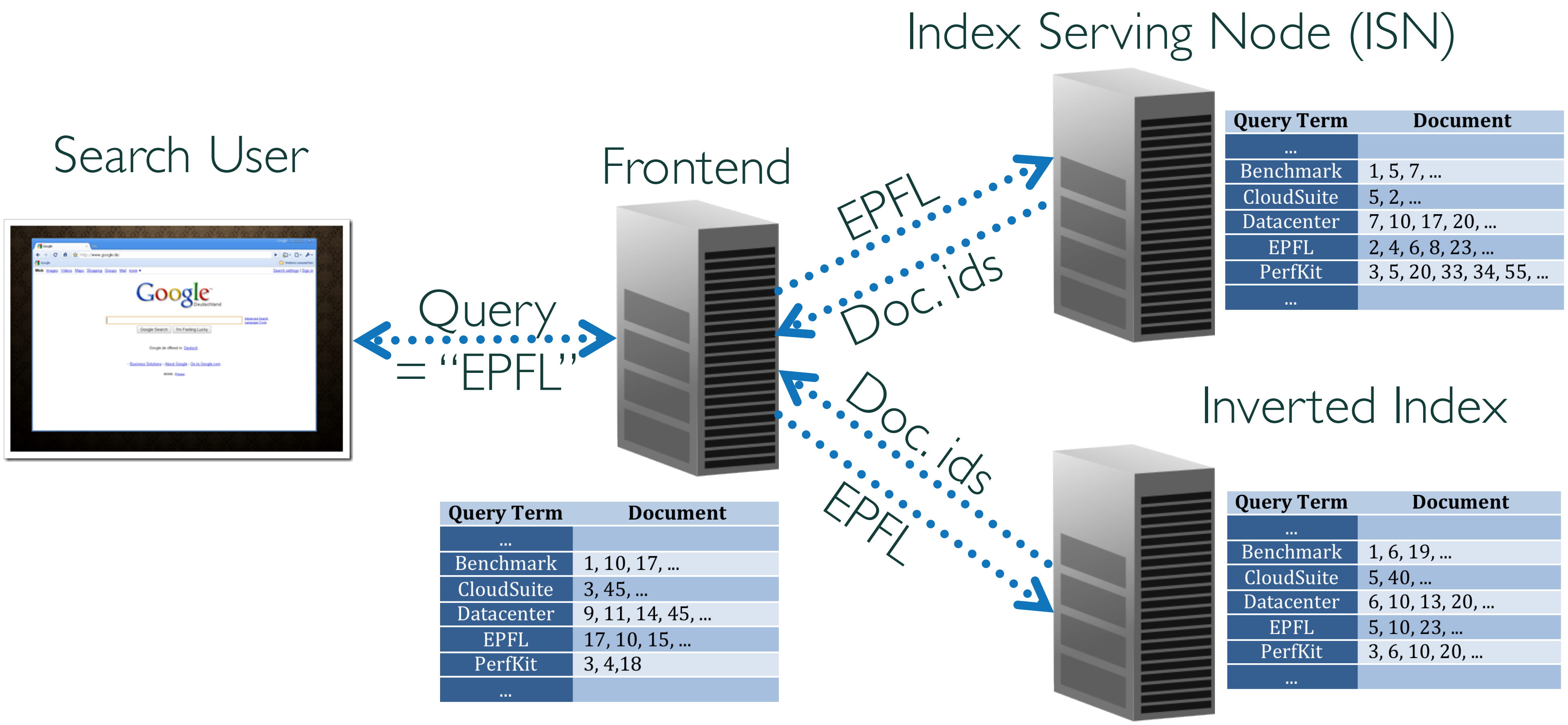


Query Term	Document
...	
Benchmark	1, 10, 17, ...
CloudSuite	3, 45, ...
Datacenter	9, 11, 14, 45, ...
EPFL	17, 10, 15, ...
PerfKit	3, 4,18
...	

Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

Web Search Operation



Index Serving Node (ISN)

Search User

Frontend

Query = "EPFL"

EPFL

Doc. ids

Doc. ids

EPFL

Inverted Index

Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Query Term	Document
...	
Benchmark	1, 10, 17, ...
CloudSuite	3, 45, ...
Datacenter	9, 11, 14, 45, ...
EPFL	17, 10, 15, ...
PerfKit	3, 4,18
...	

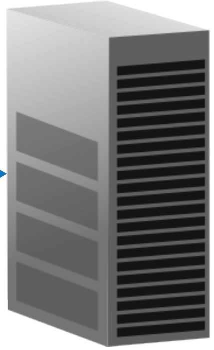
Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

Web Search Operation

Search User

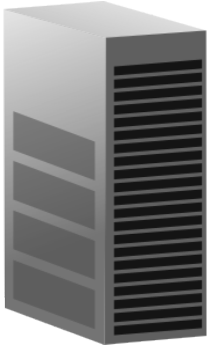
Query = "EPFL"

Frontend



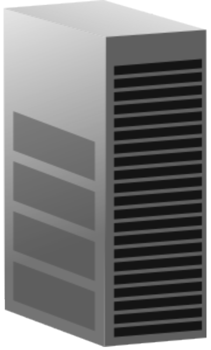
Query Term	Document
...	
Benchmark	1, 10, 17, ...
CloudSuite	3, 45, ...
Datacenter	9, 11, 14, 45, ...
EPFL	17, 10, 15, ...
PerfKit	3, 4,18
...	

Index Serving Node (ISN)



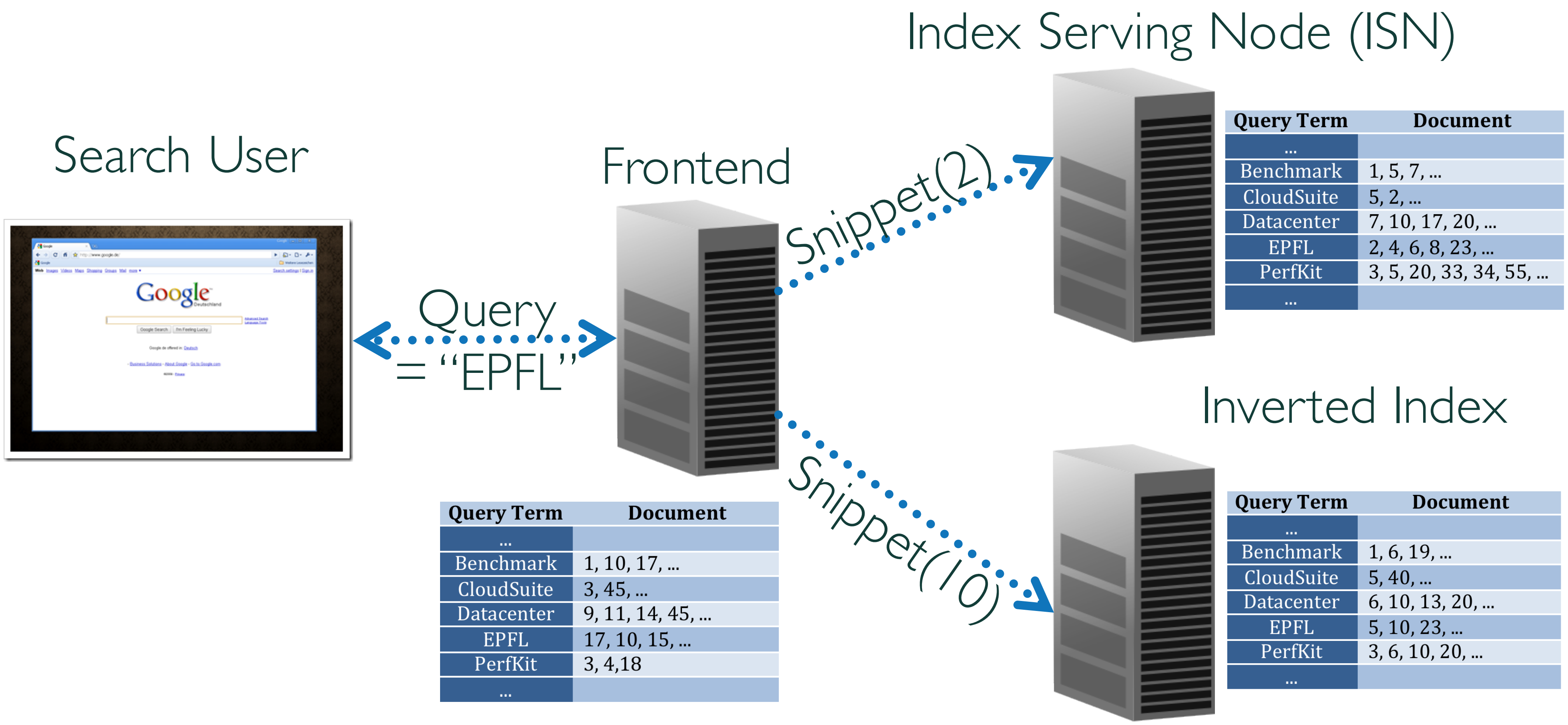
Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Inverted Index



Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

Web Search Operation

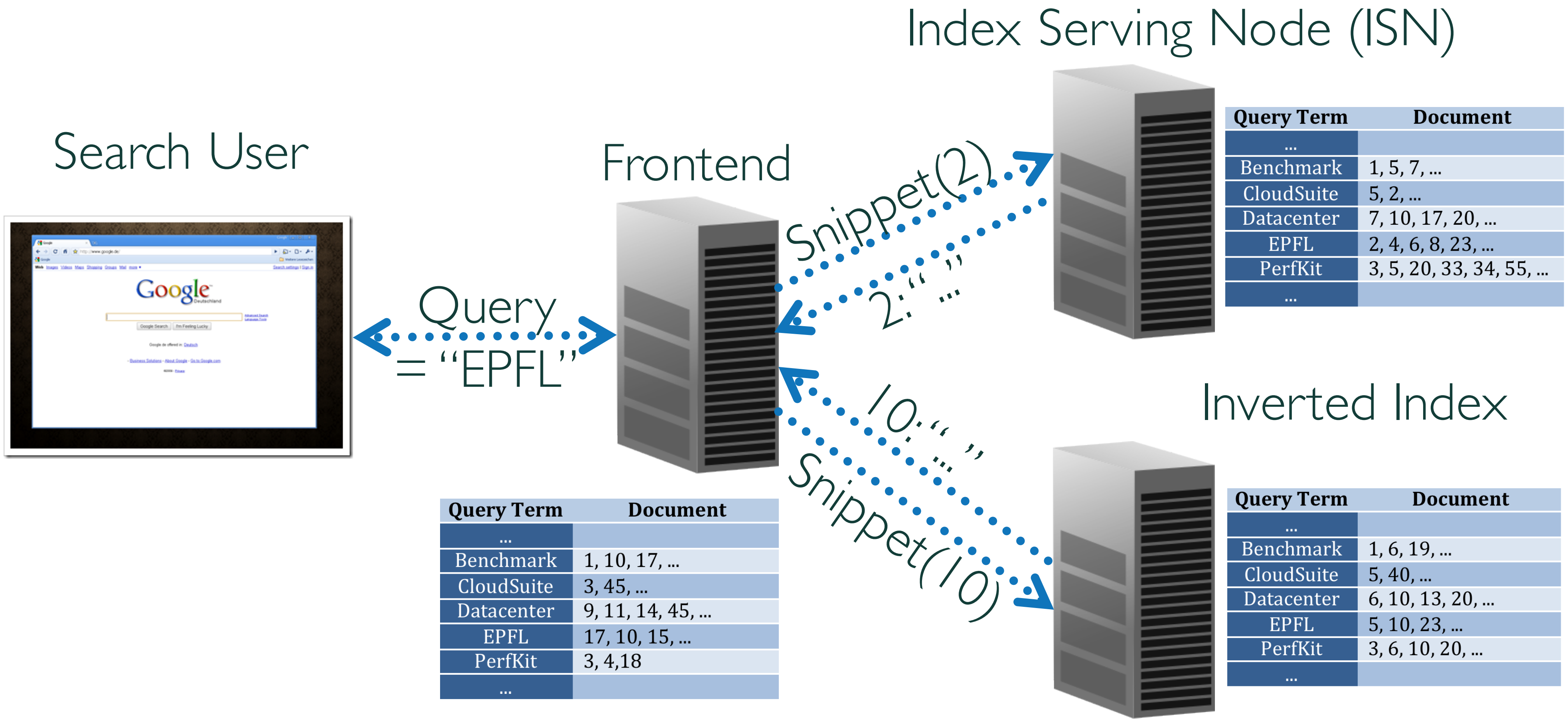


Query Term	Document
…	
Benchmark	1, 10, 17, …
CloudSuite	3, 45, …
Datacenter	9, 11, 14, 45, …
EPFL	17, 10, 15, …
PerfKit	3, 4,18
…	

Query Term	Document
…	
Benchmark	1, 5, 7, …
CloudSuite	5, 2, …
Datacenter	7, 10, 17, 20, …
EPFL	2, 4, 6, 8, 23, …
PerfKit	3, 5, 20, 33, 34, 55, …
…	

Query Term	Document
…	
Benchmark	1, 6, 19, …
CloudSuite	5, 40, …
Datacenter	6, 10, 13, 20, …
EPFL	5, 10, 23, …
PerfKit	3, 6, 10, 20, …
…	

Web Search Operation



Search User

Query = “EPFL”

Frontend

Query Term	Document
…	
Benchmark	1, 10, 17, …
CloudSuite	3, 45, …
Datacenter	9, 11, 14, 45, …
EPFL	17, 10, 15, …
PerfKit	3, 4,18
…	

Snippet(2)

2: “ … ”

10: “ … ”

Snippet(10)

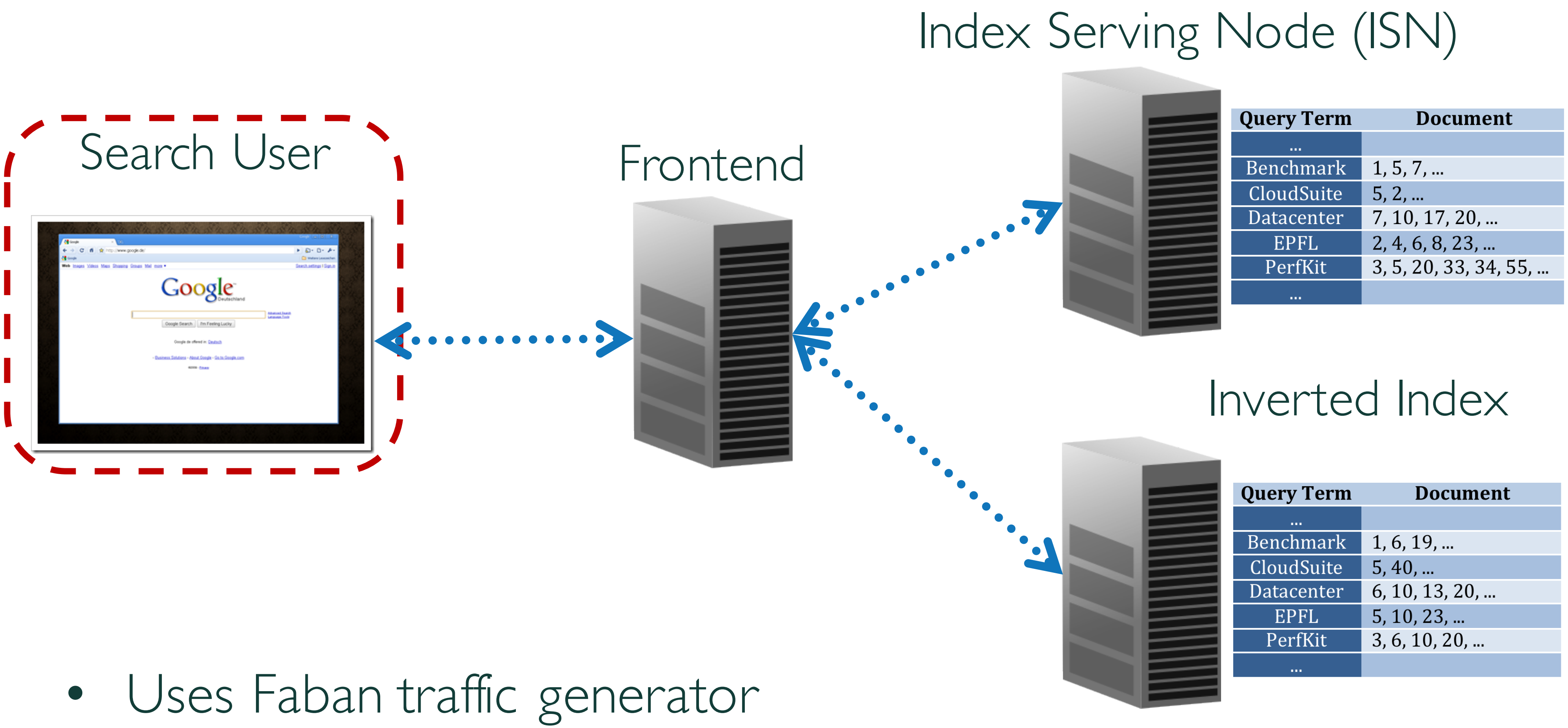
Index Serving Node (ISN)

Query Term	Document
…	
Benchmark	1, 5, 7, …
CloudSuite	5, 2, …
Datacenter	7, 10, 17, 20, …
EPFL	2, 4, 6, 8, 23, …
PerfKit	3, 5, 20, 33, 34, 55, …
…	

Inverted Index

Query Term	Document
…	
Benchmark	1, 6, 19, …
CloudSuite	5, 40, …
Datacenter	6, 10, 13, 20, …
EPFL	5, 10, 23, …
PerfKit	3, 6, 10, 20, …
…	

Web Search Operation

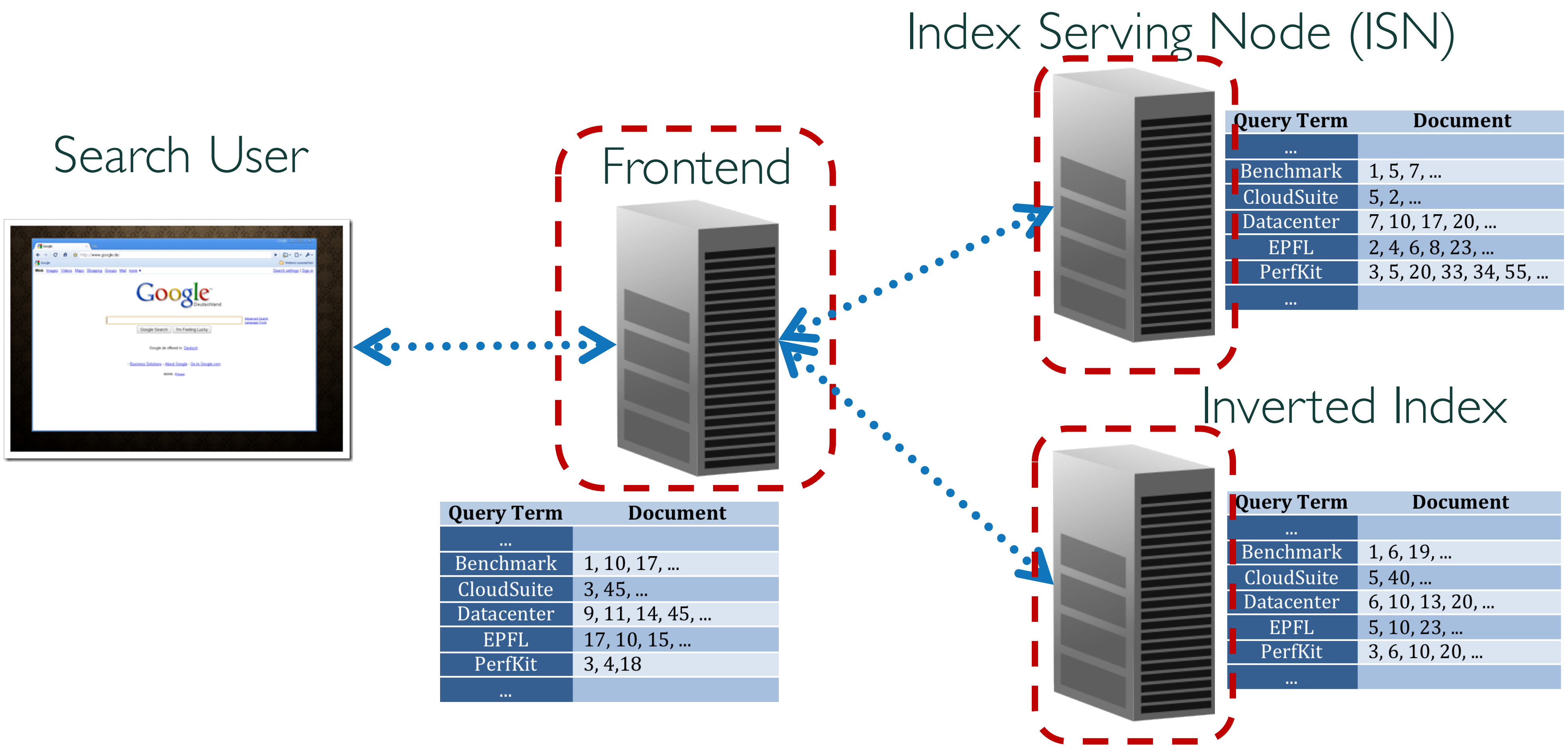


Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

- Uses Faban traffic generator
- Flexible request mixes
  - # terms per request from published surveys
  - Terms extracted from the crawled dataset

Web Search Operation



Index Serving Node (ISN)

Search User

Frontend

Inverted Index

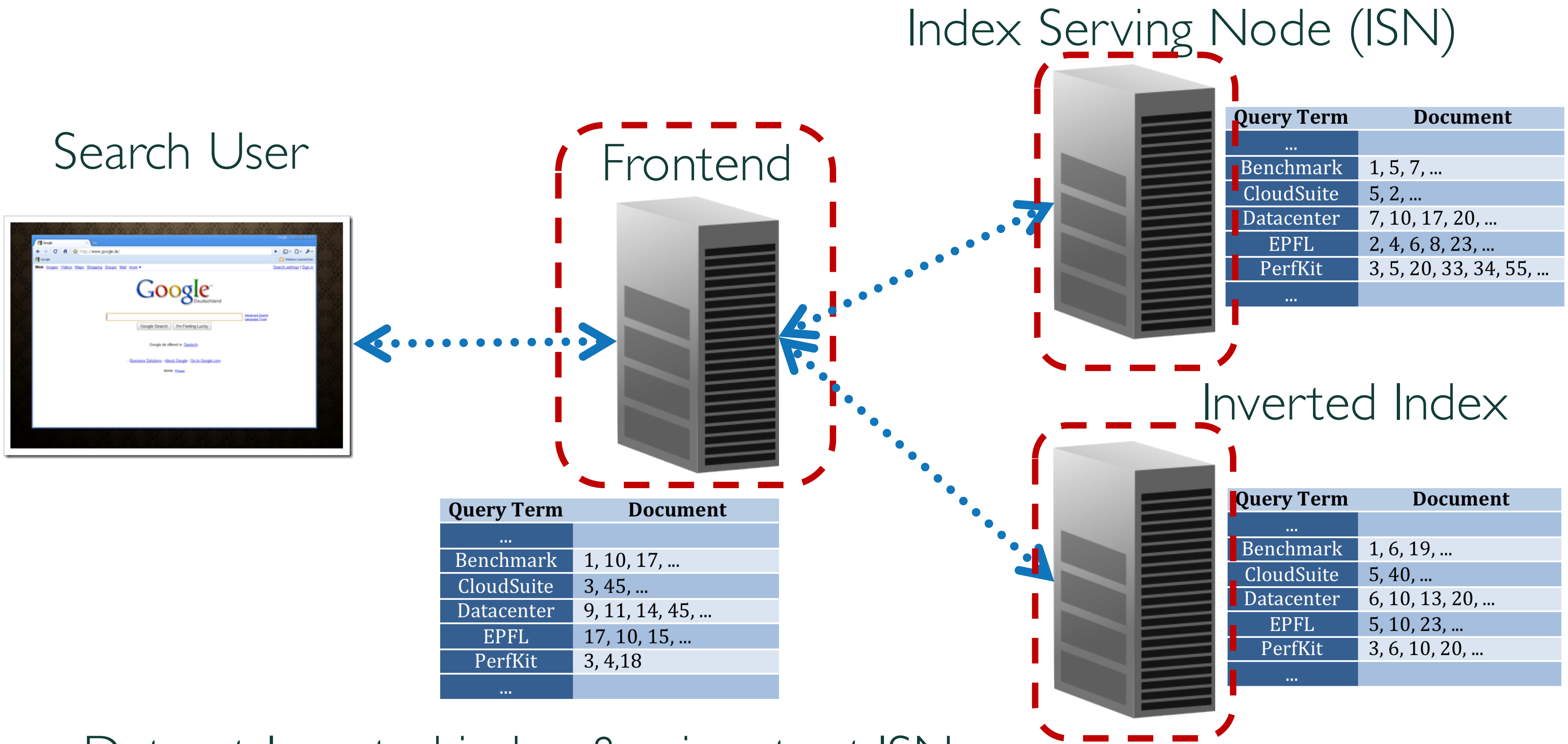
Query Term	Document
…	
Benchmark	1, 10, 17, …
CloudSuite	3, 45, …
Datacenter	9, 11, 14, 45, …
EPFL	17, 10, 15, …
PerfKit	3, 4,18
…	

Query Term	Document
…	
Benchmark	1, 5, 7, …
CloudSuite	5, 2, …
Datacenter	7, 10, 17, 20, …
EPFL	2, 4, 6, 8, 23, …
PerfKit	3, 5, 20, 33, 34, 55, …
…	

Query Term	Document
…	
Benchmark	1, 6, 19, …
CloudSuite	5, 40, …
Datacenter	6, 10, 13, 20, …
EPFL	5, 10, 23, …
PerfKit	3, 6, 10, 20, …
…	

- Apache Solr search engine for ISNs

Web Search Operation



Query Term	Document
...	
Benchmark	1, 10, 17, ...
CloudSuite	3, 45, ...
Datacenter	9, 11, 14, 45, ...
EPFL	17, 10, 15, ...
PerfKit	3, 4,18
...	

Query Term	Document
...	
Benchmark	1, 5, 7, ...
CloudSuite	5, 2, ...
Datacenter	7, 10, 17, 20, ...
EPFL	2, 4, 6, 8, 23, ...
PerfKit	3, 5, 20, 33, 34, 55, ...
...	

Query Term	Document
...	
Benchmark	1, 6, 19, ...
CloudSuite	5, 40, ...
Datacenter	6, 10, 13, 20, ...
EPFL	5, 10, 23, ...
PerfKit	3, 6, 10, 20, ...
...	

- Dataset: Inverted index & snippets at ISN
  - Generated by crawling public web (Apache Nutch)
  - Data at ISN must be memory resident
- Performance metrics: search ops/sec, N-th pct latency -

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Web Serving

- Key to all internet-based services

f amazon ebay Hotwire.com Fly. Sleep. Drive. Cheap. CNN

- All services are accessed through web servers

Apache LIGHTTPD fly light. NGINX

- Various technologies construct web content
  - HTML, PHP, JavaScript, Ruby

Web Serving Operation

Client

Web Server

Cache Server

Database Server

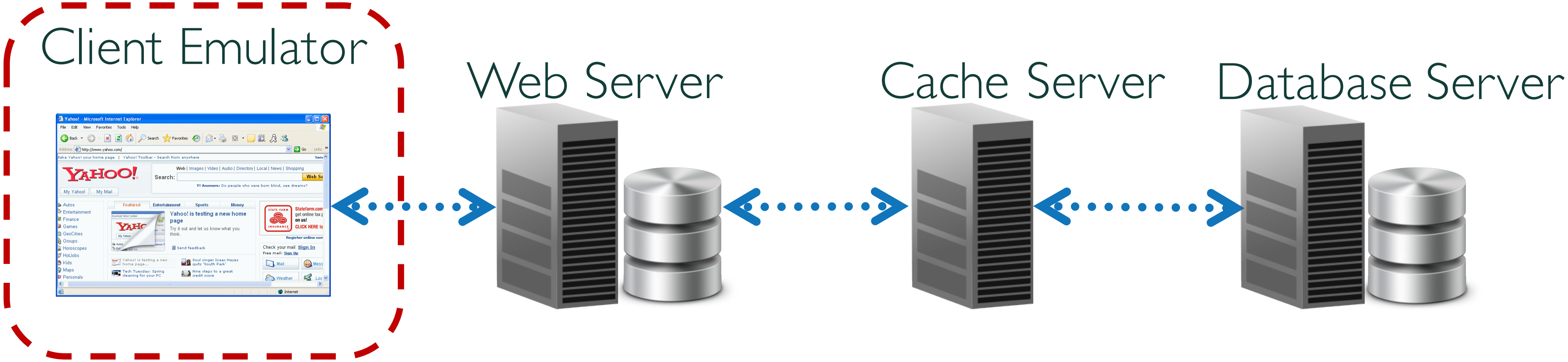
GET()

POST()

Query

Query

Web Serving Benchmark



Client Emulator

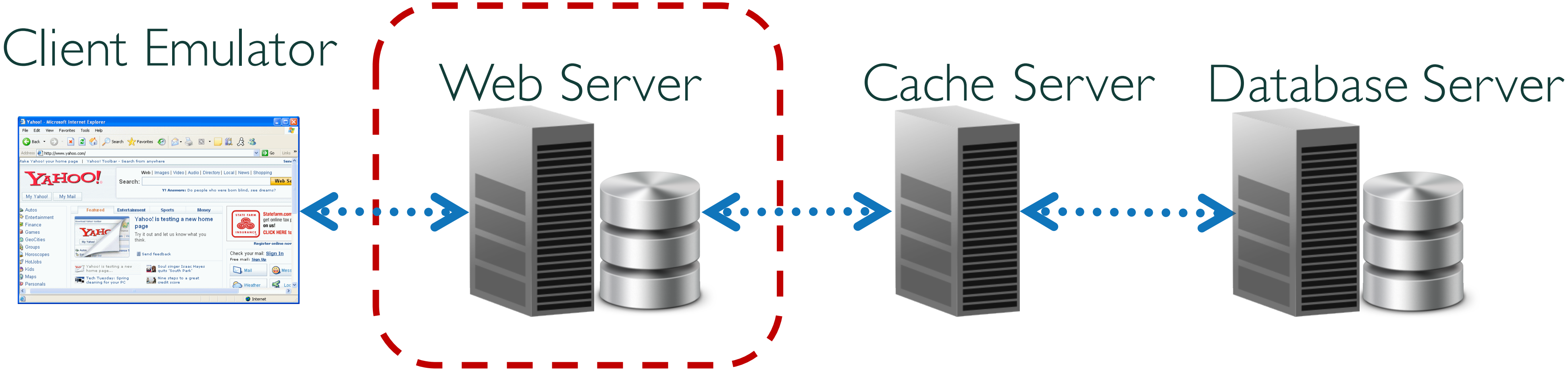
Web Server

Cache Server

Database Server

- Faban traffic generator
- Pre-configured page transition matrix (Elgg)

Web Serving Benchmark



Client Emulator

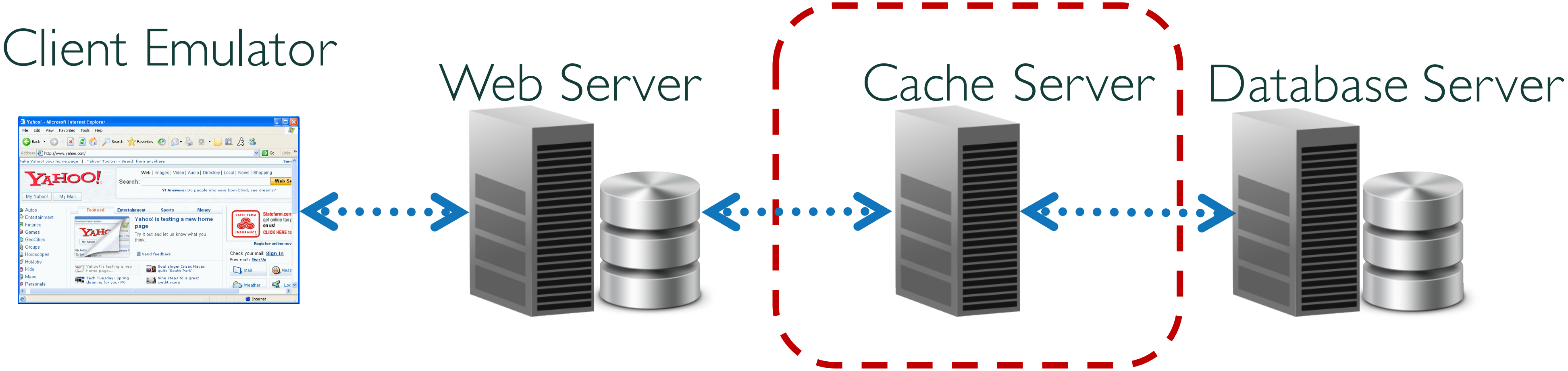
Web Server

Cache Server

Database Server

- Web server (Nginx)
- Application server (PHP)
  - Serves a social network engine (Elgg)

Web Serving Benchmark



Client Emulator

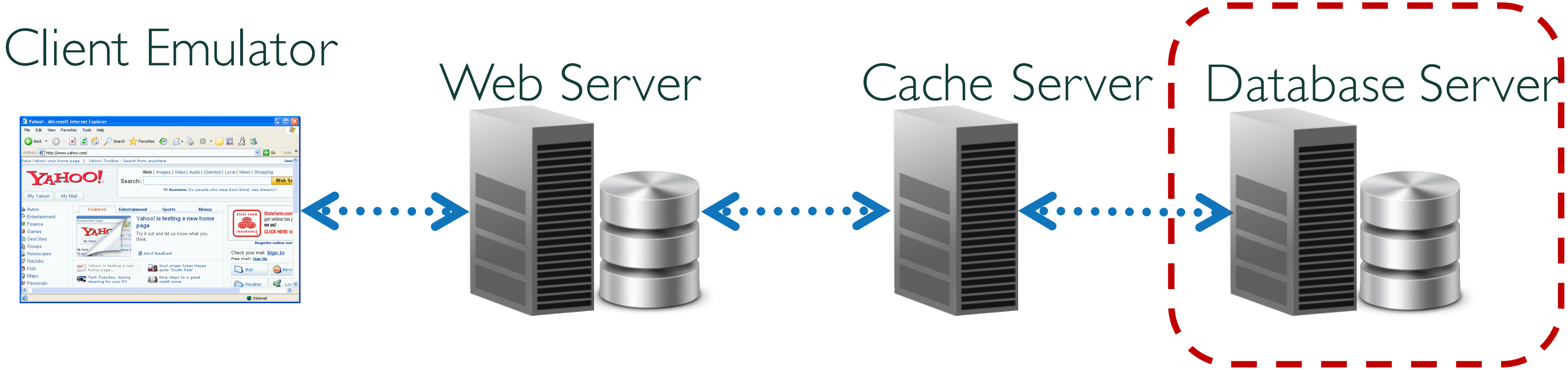
Web Server

Cache Server

Database Server

- Cache server (Memcached)

Web Serving Benchmark



Client Emulator

Web Server

Cache Server

Database Server

- Database server (MySQL)
- Performance metrics:
  # pages/second served, N-th pct latency

CloudSuite 3.0 Benchmarks

- Offline (Analytics)
  - Data Analytics
  - In-Memory Analytics
  - Graph Analytics

- Online
  - Data Caching
  - Data Serving
  - Media Streaming
  - Web Search
  - Web Serving

Future directions

- New workload: Intelligent Personal Assistants (IPAs)
  - Examples: Siri, Google Now, Alexa
- Workload overview:
  - System is queried with a question (image or sound file)
  - Applies ML techniques to convert image or sound to text
  - Text is used to query a knowledge graph
  - Answer is returned to user

Download CloudSuite 3.0

cloudsuite.ch

Hands-on Session:
CloudSuite 3.0 on Real Hardware

Alexandros Daglis

Demo Session: CloudSuite 3.0 Full-System Simulation

Javier Picorel

Software Simulation

Allows for fast & easy evaluation of a design

- Minimal cost, simulator runs on your desktop
- Reuse components, don't implement everything

Enables various benchmarks (e.g., SPEC, CloudSuite)

- Can execute real applications
- Can simulate thousands of disks
- Can simulate very fast networks

What are the simulation requirements?

CloudSuite Simulation Requirements (I)

CloudSuite Benchmarks:

- Multi-threaded, multi-processor
- Data-intensive
- Multi-tier

- Exercise OS and I/O extensively
- OS and I/O are first-order performance determinants

Need full-system simulation

CloudSuite Simulation Requirements (II)

Server architectures:

- Many-core processors
- Multiple memory controllers and memory chips
- Interconnects, cache hierarchies, …
- Custom peripherals

- Servers are complex hardware stacks
- Interaction between layers determines performance

Need detailed cycle-accurate simulation

CloudSuite Simulation Requirements (III)

CloudSuite benchmarks:

- Seconds of execution (trillions of instructions)

Full-system and cycle-accurate simulation:

- 1M slowdown vs. real hardware

- Long benchmarks and slow simulators
- Years of simulation time per experiment

Need low simulation turnaround times

Simulation Stack: QFlex

- Functional Full-System Simulation: **QEMU**
- Detailed Microarchitectural Simulation: **Flexus**
- Fast Simulation: **Statistical sampling**

Full-System Simulation with QEMU

Full-System Simulation Requirements

Full-system functional simulator must support:

- Privileged-mode ISA
- I/O devices
- Networks of systems
- Saving/restoring architecturally-visible state

QEMU provides these capabilities

QEMU Configuration & API

- Configuration file defines system components
  - Motherboard, CPUs, memory, I/O devices
- Extension to QEMU provides interface to simulation
  - Start and stop simulation
  - Access to target system's architecturally-visible state
  - Callback system under certain architectural events
  - Save and restore target system's architecturally-visible state

QEMU Interface

QEMU does not provide timing details

- But provides an architectural interface
- Allows a user module to take control over timing

QEMU provides Flexus with instructions

Flexus controls instruction flow from QEMU

- Flexus requests instructions from QEMU
- Executes received instructions in cycle-accurate mode

Detailed Microarchitectural Simulation with Flexus

Main Idea

Use existing machine emulator (QEMU)

- Handles BIOS (booting, I/O, interrupt routing, . . .)

Build a "plugin" architectural model simulator

- Fast – read system's state from QEMU
- Detailed – interact with and throttle QEMU

Developing with Flexus

- Flexus philosophy
- Fundamental abstractions
- Important support libraries
- Simulators and components in Flexus

Flexus philosophy

Component-based design

- Compose simulators from encapsulated components

Software-centric framework

- Flexus abstractions are not tied to hardware

Cycle-driven execution model

- Components receive “clock-tick” signal every cycle

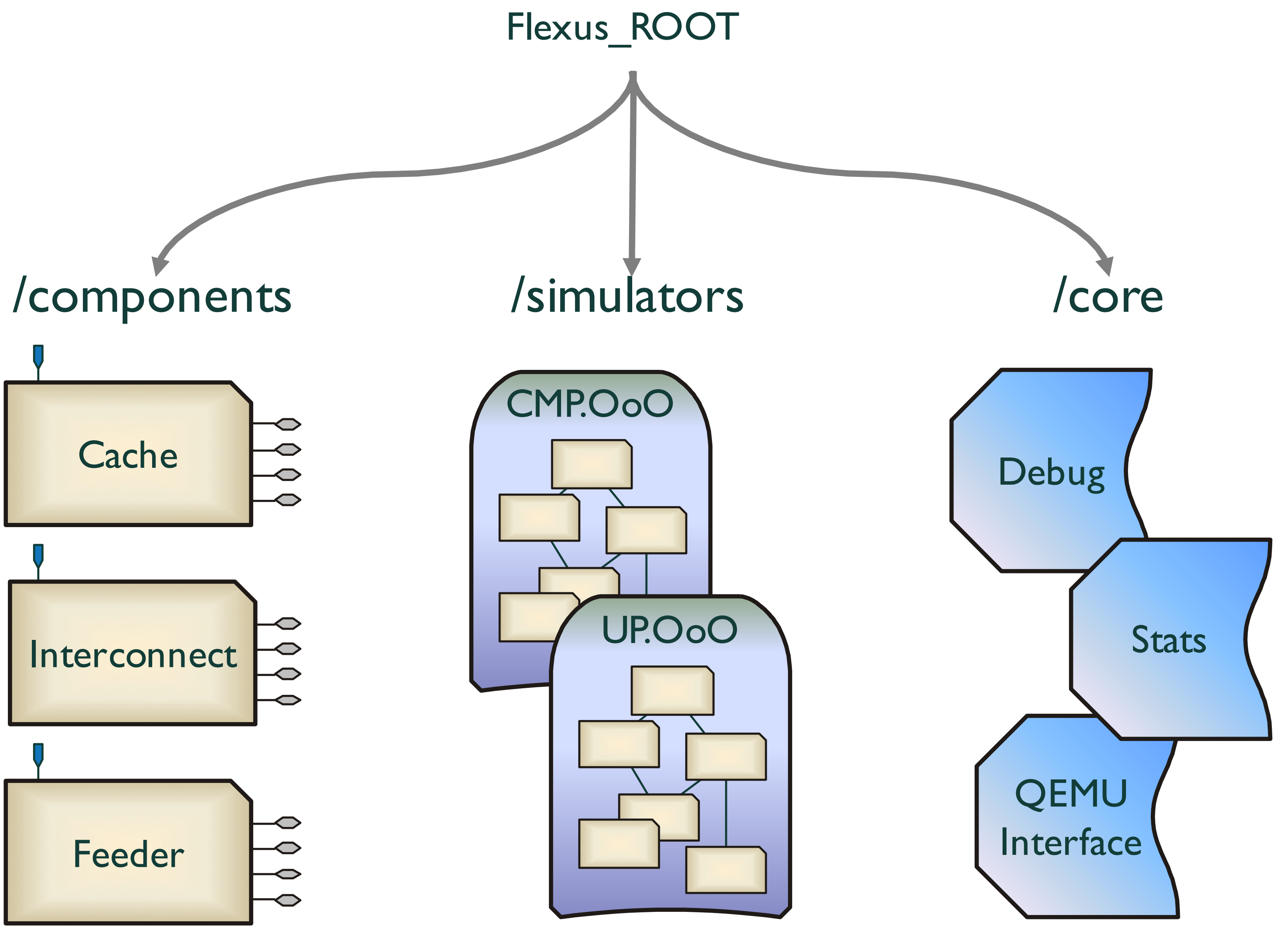
SimFlex methodology

- Designed-in fast-forwarding, checkpointing, statistics

Developing with Flexus

- Flexus philosophy
- **Fundamental abstractions**
- Important support libraries
- Simulators and components in Flexus

Flexus organization



Fundamental abstractions

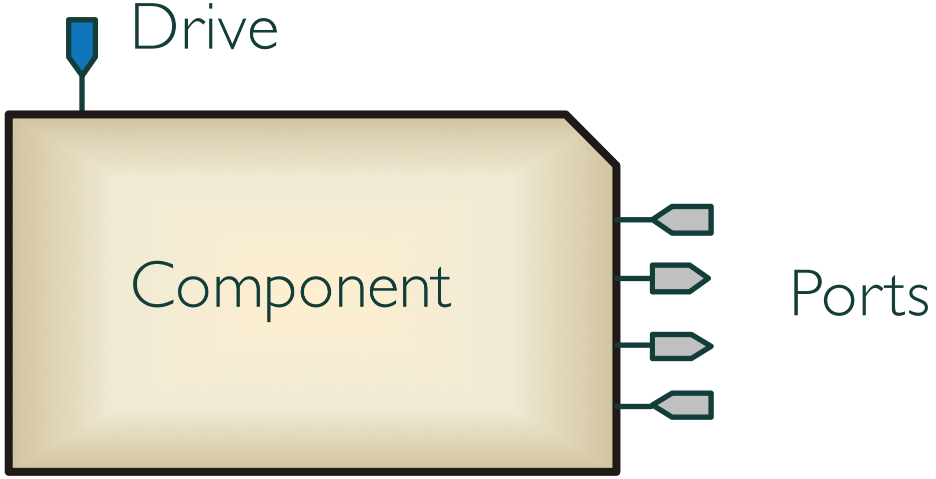
Component

- Component interface
    - Specifies data and control entry points
- Component parameters
    - Configuration settings available in configuration file

Simulator

- Wiring
    - Specifies which components and how to connect
    - Specifies default component parameter settings

Component interface

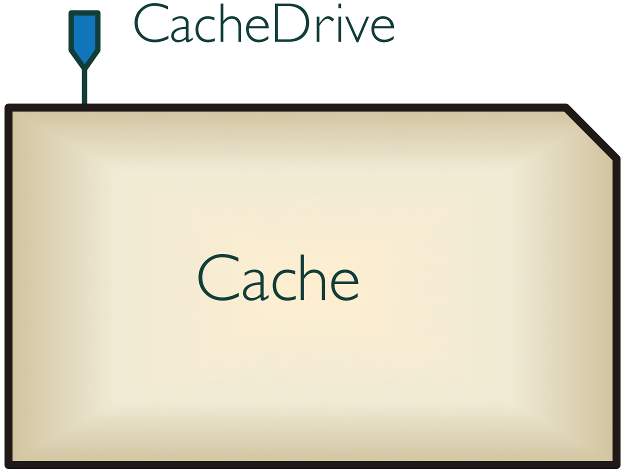


Component interface (terminology inspired by Asim [Emer 02])

- Drive: "clock-tick" control entry point to component
- Port: specifies data flow between components

Components w/ same ports are interchangeable

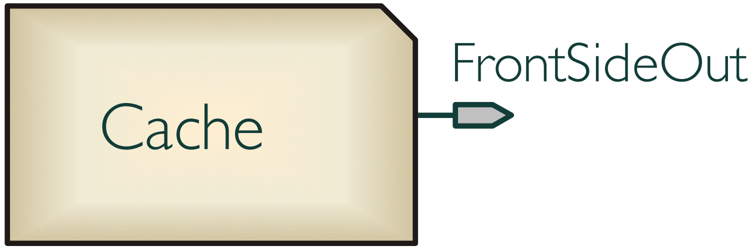
Abstractions: Drive



```
COMPONENT_INTERFACE(
    …
    DRIVE (  Name )
    …
);
```

- Control entry-point
- Function called once per cycle

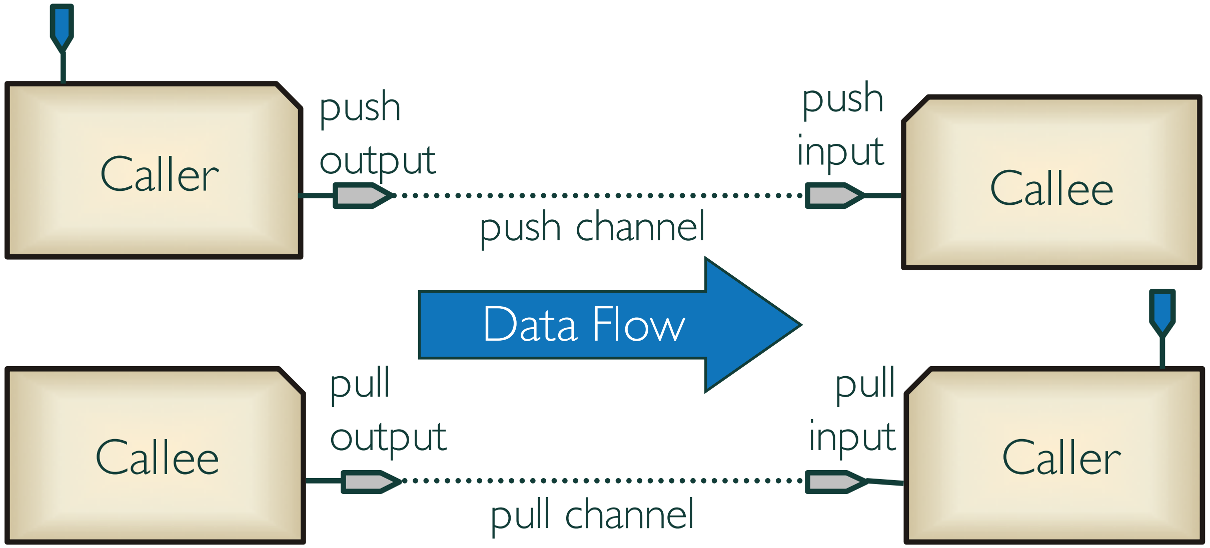
Abstractions: Port



```
COMPONENT_INTERFACE(
      …
PORT (Type,Payload,Name)
      …
);
```

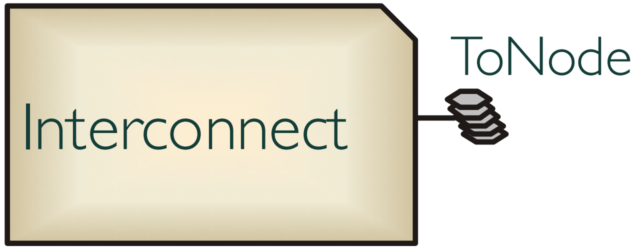
- Data exchange between components
- Ports connected together in simulator wiring

Types of ports and channels



- Type - direction of data and control flow
  - Control flow: Push vs. Pull
  - Data flow: Input vs. Output
- Payload - arbitrary C++ data type
- Type and payload must match to connect ports
- Availability - caller must check if callee is ready

Port and component arrays



```
COMPONENT_INTERFACE(
    ...
DYNAMIC_PORT_ARRAY(...)
    ...
);
```

- 1-to-n and n-to-n connections
  - E.g., 1 interconnect -> n network interfaces
- Array dimensions can be dynamic

Example code using a port

```
SenderComponent.cpp
 void someFunction() {
     Message msg;
     if ( FLEXUS_CHANNEL(Out).available() )
 {
            FLEXUS_CHANNEL(Out) << msg;
     }
 }
ReceiverComponent.cpp
 bool available( interface::In ) {
     return true;   }

 void push( interface::In, Message & msg)
     { … }
```

Configuring components

- Configurable settings associated with component
  - Declared in component specification
  - Can be std::string, int, long, long long, float, double, enum
  - Declaration:

```
   PARAMETER( BufferSize, int, “L2 Buffer
size", “bsize", 64 )
```

  - Use: `cfg.bsize`

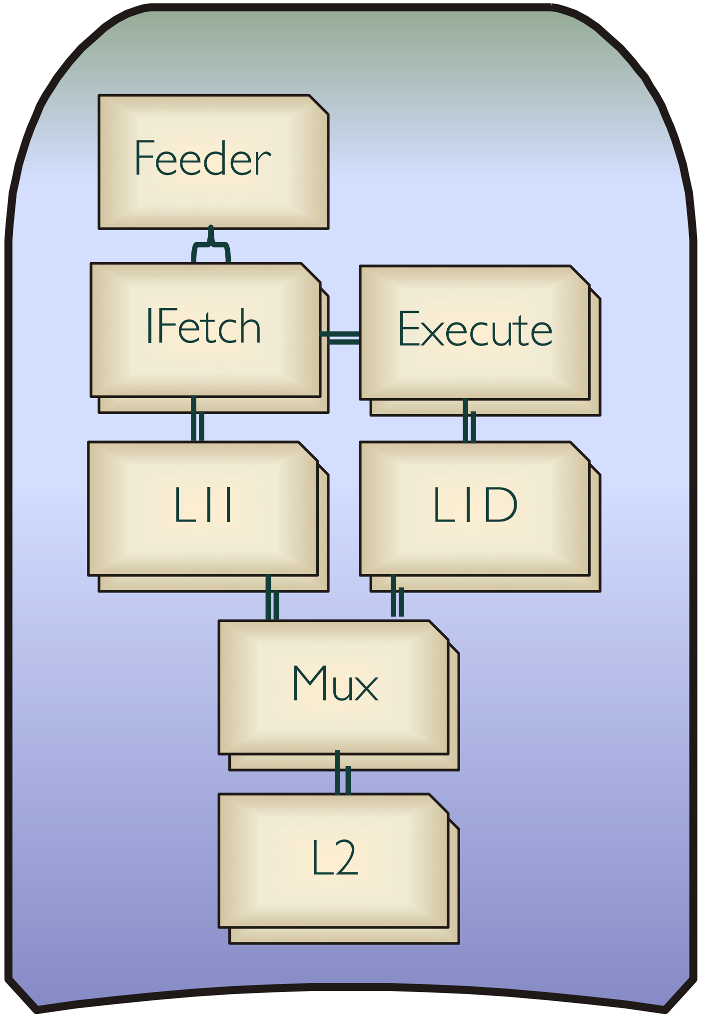
Simulator wiring

simulators/name/Makefile.name

- List components for link
- Indicate target support

simulators/name/wiring.cpp

1. Include interfaces
2. Declare configurations
3. Instantiate components
4. Wire ports together
5. List order of drives



Developing with Flexus

- Flexus philosophy
- Fundamental abstractions
- **Important support libraries**
- Simulators and components in Flexus

Critical support libraries in /core

- Statistics support library
  - Record results for use with `stat-manager`

- Debug library
  - Control and view Flexus debug messages

Statistics support library

- Implements all the statistics you need
  - Histograms
  - Unique counters
  - Instance counters
  - etc.

- Example:

```
Stat::StatCounter  myCounter(
statName() + “-count” );
++ myCounter;
```

A typical debug statement

```
DBG_(Iface,
    Comp(*this),
    AddCategory( Cache ),
     (   << "Received on
  FrontSideIn[0](Request):  "
         << *(aMessage[MemoryMessageTag])
      ),
Addr(aMessage[MemoryMessageTag]->address())
);
```

Severity level

Associate with this component

Put this in “Cache” category

Text of the debug message

Add an address field for filtering

Debug severity levels

1.	Tmp	temporary messages (cause warning)
2.	Crit	critical errors
3.	Dev	infrequent messages, e.g., progress
4.	Trace	component defined – typically tracing
5.	Iface	all inputs and outputs of a component
6.	Verb	verbose output from OoO core
7.	Vverb	very verbose output of internals

Compile time

- `make target-severity`
- (e.g., `make CMP.Trace-iface`)

Developing with Flexus

- Flexus philosophy
- Fundamental abstractions
- Important support libraries
- **Simulators and components in Flexus**

Simulators in Flexus

Trace simulation

- Every instruction executes in a single cycle
- Fast Multi-level memory hierarchy
- Fast Two-level branch prediction

Timing (cycle-accurate) simulation

- "Cycle-by-cycle" execution of on-chip components:
    - Cores
    - NoC
    - Memory hierarchy
    - …

Memory hierarchy

Allows for high MLP

- Non-blocking, pipelined accesses
- Hit-under-miss within set

Coherence protocol support

- MESI and MOESI coherence protocols
- Non-inclusive cache hierarchy
- Supports "Downgrade" and "Invalidate" messages
- Request and snoop virtual channels for progress guarantees

DRAMSim 2.0 integrated for low-level DRAM simulation

Out-of-order execution

- Timing-first simulation approach [Mauer'02]
  - OoO components interpret ARM (32/64 bits) ISA
  - Flexus validates its results with QEMU

- Idealized OoO to maximize memory pressure
  - Decoupled front-end
  - Precise squash & re-execution
  - Configurable ROB, SB; dispatch, retire rates

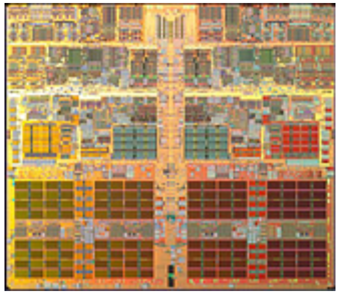
- Memory consistency models (SC, TSO, RMO)

Fast Simulation with Statistical Sampling

Simulation Speed Challenges

- Longer benchmarks
  - CloudSuite: Trillions of instructions per benchmark

- Slower simulators
  - Full-system simulation: 1000× slower than real hardware
  - Cycle-accurate simulation: 1000x slower than full-system simulation



- Multiprocessor systems
  - CMP: 2x cores every processor generation

1,000,000 × slowdown vs. HW → years per experiment

Full-system simulation is slow

- Simulation slowdown per CPU core

Real HW:	~ 2 GIPS	1 s
QEMU:	~ 30 MIPS	66 s
Flexus, no timing:	~ 900 KIPS	37 m
Flexus, OoO:	~ 24 KIPS	23 h

2 years to simulate 10 seconds of a 64-core workload!

Statistical Sampling

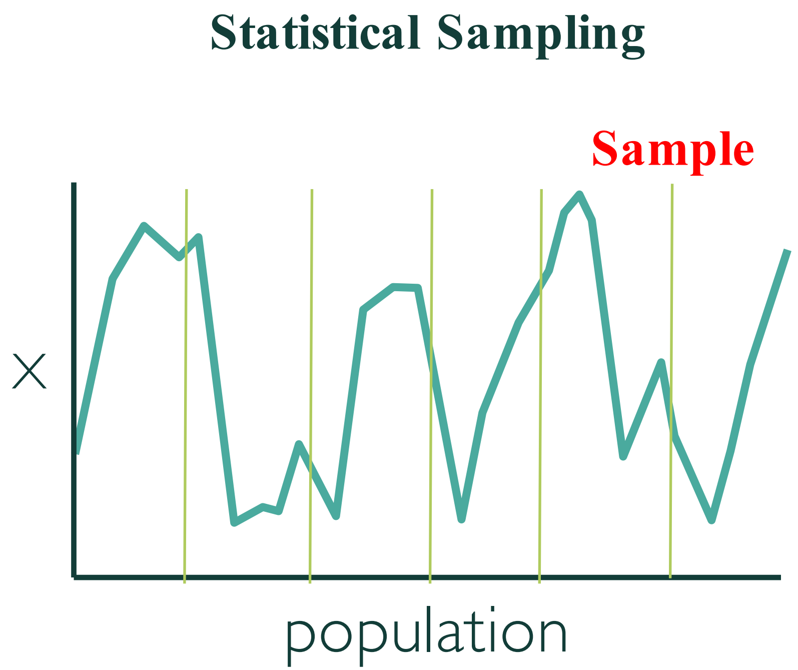
Random selection of population

- E.g., 3000 out of 300 million

Predict the behavior based on the selected sample

Features:

- High accuracy
- Simple
- Strong mathematical foundation



Power of a small part to predict behavior of a whole

Simulation Speedup (I)

Measure in detailed a few parts of the execution

- From few years to a month

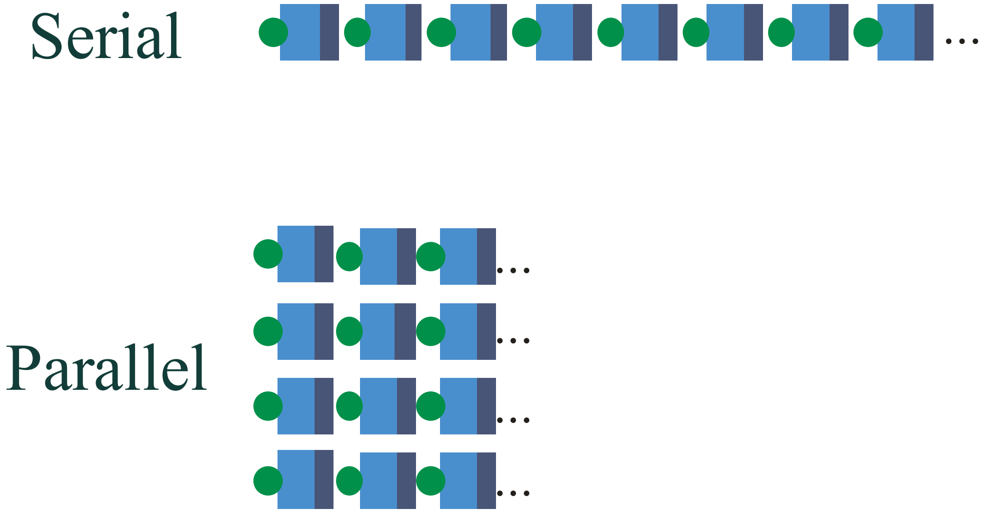
Store functional warming

- Saves us a few more days

From a few years to less than a month…

Simulation Speedup (II)

Store functional warming and **parallelize**



From less than a month to a few hours!

QFlex Status

QFlex project still an on-going effort…stay tuned!

- **QEMU** Function Full-System Simulation ✓
- **QFlex** Trace simulation ✓
- **QFlex** cycle-accurate simulation

- Statistical **sampling** for fast simulation

DEMO Session: QFlex Trace Simulator

DEMO Session: Overview

"Add a next-line prefetcher with configurable prefetch degree length"

Steps:

1. QFlex structure overview
2. Create the new component
3. Create a new simulator for the new component
4. Run the new simulator and the baseline
5. Extract and compare results

Send us an email to test our simulator!

Thank You!

PARALLEL SYSTEMS
ARCHITECTURE LAB

For more information please visit us at
parsa.epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE